

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-181908

(43)Date of publication of application : 11.07.1997

(51)Int.Cl.

H04N 1/387

G06T 1/00

G09G 5/06

G09G 5/36

H04N 1/60

H04N 1/46

(21)Application number : 08-255499

(71)Applicant : CANON INF SYST INC

(22)Date of filing : 05.09.1996

(72)Inventor : KOHLER TIMOTHY L

HUI JONATHAN Y

(30)Priority

Priority number : 95 529111

Priority date : 15.09.1995

Priority country : US

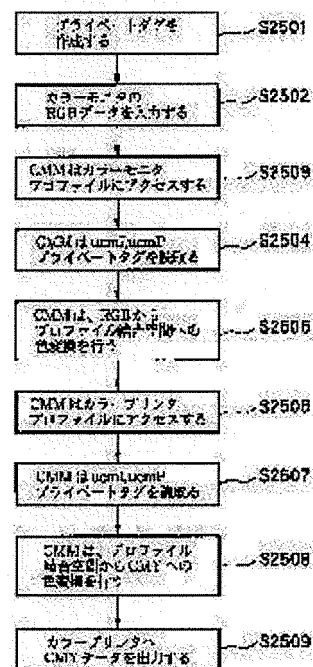
(54) COLOR IMAGE PROCESSOR AND ITS METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To allow the processor to apply color conversion arithmetic operation to received image data by using 1st and 2nd tags in place of prescribed series of color transformation operation stored in a public tag of a profile format.

SOLUTION: This processor includes a color image data input step S2502, a discrimination step S52503 storing a 1st tag that stored override information invalidating prescribed series color transformation operation and a 2nd tag storing color conversion arithmetic operation data accessible via a hierarchical storage structure, and a discrimination step S5206 discrimination whether or not the 2nd tag is to be accessed based on the override information of the 1st tag, and also a read step S5207

reading the color transformation data according to a pointer of a hierarchical storage structure



to access the color transformation data of the 2nd tag and a processing step S5208 applying color transformation operation to the input image data according to the read color transformation data.

LEGAL STATUS

[Date of request for examination] 12.02.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3902816

[Date of registration] 12.01.2007

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-181908

(43) 公開日 平成9年(1997)7月11日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	FI	技術表示箇所
H04N 1/387			H04N 1/387	
G06T 1/00		9377-5H	G09G 5/06	
G09G 5/06		9377-5H	5/36	520A
	520		G06F 15/66	310
H04N 1/60			H04N 1/40	D

審査請求 未請求 請求項の数34 OL 外国語出願 (全105頁) 最終頁に続く

(21) 出願番号 特願平8-255499

(22) 出願日 平成8年(1996)9月5日

(31) 優先権主張番号 08/529111

(32) 優先日 1995年9月15日

(33) 優先権主張国 米国 (US)

(71) 出願人 592208172

キヤノン インフォメーション システムズ インク.

Canon Information Systems, Inc.

アメリカ合衆国 カリフォルニア州

92626, コスタ メサ, ブルマン ストリート 3188

(72) 発明者 ティモシ エル. コーラ

アメリカ合衆国 カリフォルニア州

94043, マウンテン ビュー, #16, ロック ストリート 2434

(74) 代理人 弁理士 大塚 康德 (外1名)

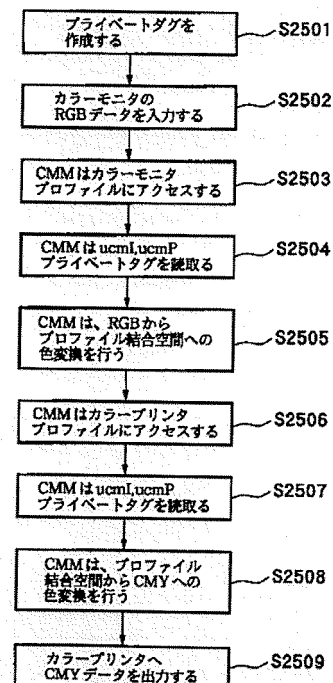
最終頁に続く

(54) 【発明の名称】 カラー画像処理装置およびその方法

(57) 【要約】

【課題】 システムは、プロファイルフォーマットのパブリックタグに格納された所定の一連の色変換演算に代わり、第一のタグおよび第二のタグを使用して、入力画像データに色変換演算を施す。

【解決手段】 システムは、カラー画像データの入力ステップ、所定の一連の色変換演算を無効にするオーバーライド情報を格納する第一のタグと、階層記憶構造を介してアクセス可能な色変換演算データを格納する第二のタグとを格納する格納ステップ、第一のタグのオーバーライド情報に基づき、第二のタグにアクセスすべきか否かを判定する判定ステップを含む。さらに、第二のタグにアクセスすべきと判定した場合、第二のタグの色変換演算データにアクセスするための階層記憶構造のポインタに従い、色変換演算データを読み取る読取ステップ、読取った色変換演算データに基づき、入力画像データに色変換演算を施す処理ステップを含む。



【特許請求の範囲】

【請求項1】 プロファイルフォーマットのパブリックタグに格納された所定の一連の色変換演算の代わりに、第一のタグおよび第二のタグを使用して、入力画像データに色変換演算を施すカラー画像処理装置であって、カラー画像データを入力する入力手段と、前記所定の一連の色変換演算を無効にするオーバーライド情報を格納する前記第一のタグと、階層記憶構造を介してアクセス可能な色変換演算データを格納する前記第二のタグとを格納するメモリと、前記第一のタグのオーバーライド情報に基づき、前記第二のタグにアクセスすべきか否かを判定する判定手段と、前記判定手段により前記第二のタグにアクセスすべきであると判定された場合、前記第二のタグの色変換演算データにアクセスするための前記階層記憶構造のポインタに従い、前記色変換演算データを読取る読取手段と、前記読取手段により読取られた色変換演算データに基づき、前記入力画像データに色変換演算を施す処理手段とを有することを特徴とするカラー画像処理装置。

【請求項2】 前記第一のタグに格納されたオーバーライド情報は、前記第二のタグ中の色変換シーケンスのアクセスに使用され、前記読取手段は、前記オーバーライド情報に基づき、前記色変換シーケンスをアクセスすることを特徴とする請求項1に記載されたカラー画像処理装置。

【請求項3】 前記色変換シーケンスは、前記色変換演算を指すポインタを格納し、前記読取手段は、前記色変換シーケンスに格納された前記ポインタに基づき、前記色変換演算にアクセスすることを特徴とする請求項2に記載されたカラー画像処理装置。

【請求項4】 前記メモリは、前記色変換演算のリストを格納し、前記処理手段は、前記読取手段により読取られた色変換演算データに基づき、前記色変換演算のリスト中の一つにアクセスすることを特徴とする請求項1に記載されたカラー画像処理装置。

【請求項5】 前記判定手段により前記第二のタグにアクセスすべきでないと判定された場合、前記入力画像データは、前記パブリックタグに格納された前記所定の一連の色変換演算に基づき処理されることを特徴とする請求項1に記載されたカラー画像処理装置。

【請求項6】 プロファイルフォーマットのパブリックタグに格納された所定の一連の色変換演算の代わりに、第一のタグおよび第二のタグを使用して、入力画像データに色変換演算を施すカラー画像処理方法であって、カラー画像データを入力する入力ステップと、前記所定の一連の色変換演算を無効にするオーバーライド情報を格納する前記第一のタグと、階層記憶構造を介してアクセス可能な色変換演算データを格納する前記第二のタグとを格納する格納ステップと、前記第一のタグのオーバーライド情報に基づき、前記第二のタグにアクセスすべきか否かを判定する判定ステップと、前記第一のタグのオーバーライド情報に基づき、前記第二

のタグにアクセスすべきか否かを判定する判定ステップと、前記判定ステップで前記第二のタグにアクセスすべきであると判定された場合、前記第二のタグの色変換演算データにアクセスするための前記階層記憶構造のポインタに従い、前記色変換演算データを読取る読取ステップと、前記読取ステップで読取られた色変換演算データに基づき、前記入力画像データに色変換演算を施す処理ステップとを有することを特徴とするカラー画像処理方法。

【請求項7】 前記第一のタグに格納されたオーバーライド情報は、前記第二のタグ中の色変換シーケンスのアクセスに使用され、前記読取ステップは、前記オーバーライド情報に基づき、前記色変換シーケンスをアクセスすることを特徴とする請求項6に記載されたカラー画像処理装置。

【請求項8】 前記色変換シーケンスは、前記色変換演算を指すポインタを格納し、前記読取ステップは、前記色変換シーケンスに格納された前記ポインタに基づき、前記色変換演算にアクセスすることを特徴とする請求項7に記載されたカラー画像処理装置。

【請求項9】 前記格納ステップは、前記色変換演算のリストを格納し、前記処理ステップは、前記読取ステップで読取られた色変換演算データに基づき、前記色変換演算のリスト中の一つにアクセスすることを特徴とする請求項6に記載されたカラー画像処理装置。

【請求項10】 前記判定ステップで前記第二のタグにアクセスすべきでないと判定した場合、前記入力画像データは、前記パブリックタグに格納された前記所定の一連の色変換演算に基づき処理されることを特徴とする請求項6に記載されたカラー画像処理装置。

【請求項11】 コンピュータが使用可能な記憶媒体に記憶され、プロファイルフォーマットのパブリックタグに格納された所定の一連の色変換演算の代わりに、第一のタグおよび第二のタグを使用して、入力画像データに色変換演算を施すコンピュータが実行可能な処理ステップであって、カラー画像データを入力する入力ステップと、前記所定の一連の色変換演算を無効にするオーバーライド情報を格納する前記第一のタグと、階層記憶構造を介してアクセス可能な色変換演算データを格納する前記第二のタグとを格納する格納ステップと、前記第一のタグのオーバーライド情報に基づき、前記第二のタグにアクセスすべきか否かを判定する判定ステップと、前記判定ステップで前記第二のタグにアクセスすべきであると判定された場合、前記第二のタグの色変換演算データにアクセスするための前記階層記憶構造のポインタに従い、前記色変換演算データを読取る読取ステップと、

前記読取ステップで読取られた色変換演算データに基づき、前記入力画像データに色変換演算を施す処理ステップとを有することを特徴とする処理ステップ。

【請求項12】 前記第一のタグに格納されたオーバーライド情報は、前記第二のタグ中の色変換シーケンスのアクセスに使用され、前記読取ステップは、前記オーバーライド情報に基づき、前記色変換シーケンスをアクセスすることを特徴とする請求項11に記載された処理ステップ。

【請求項13】 前記色変換シーケンスは、前記色変換演算を指すポイントを格納し、前記読取ステップは、前記色変換シーケンスに格納された前記ポイントに基づき、前記色変換演算にアクセスすることを特徴とする請求項12に記載された処理ステップ。

【請求項14】 前記格納ステップは、前記色変換演算のリストを格納し、前記処理ステップは、前記読取ステップで読取られた色変換演算データに基づき、前記色変換演算のリスト中の一つにアクセスすることを特徴とする請求項11に記載された処理ステップ。

【請求項15】 前記判定ステップで前記第二のタグにアクセスすべきでないと判定した場合、前記入力画像データは、前記パブリックタグに格納された前記所定の連続の色変換演算に基づき処理されることを特徴とする請求項6に記載された処理ステップ。

【請求項16】 ある色空間から別の色空間への標準化された変換に使用される数値情報が格納され、タグが階層格納されているプロファイルフォーマットにおいて、前記タグを使用して前記標準化された変換を変更するカラー画像処理装置であって、

カラー画像データを入力する入力手段と、

(1) タグの階層の最高位にある第一のタグの所定の位置に、前記タグの階層の第二の高位にある第二のタグ中の色変換シーケンスを指す第一のポインタデータを格納し、(2) 前記タグの階層の第三の高位にある第三のタグに格納された色変換演算を指す第二のポインタを含む前記色変換シーケンスを前記第二のタグに格納し、(3) 前記色変換演算を実行するためのデータを第三のタグに格納するメモリと、

前記プロファイルフォーマットに格納されている情報に基づき、前記第一のタグにアクセスすべきか否かを判定する判定手段と、

(1) 前記第一のタグにアクセスすべき場合、前記第一のタグの前記第一のポインタの情報を読取り、前記色変換シーケンスにアクセスすべきか否かを判定し、(2) 前記色変換シーケンスにアクセスすべき場合は、前記色変換演算にアクセスすべきか否かを判定するために、前記色変換シーケンス中の前記第二のポインタを読取り、(3) 前記色変換演算にアクセスすべき場合には、前記第三のタグから色変換演算を実行するためのデータを読取る読取手段と、

前記標準化された変換の代わりに、ある色空間から別の色空間へ前記入力画像データを変換するために、前記第三のタグからのデータにより読取ったデータに基づき、前記色変換演算を行う処理手段とを有することを特徴とするカラー画像処理装置。

【請求項17】 前記色変換演算が所定の色変換演算を含む場合、前記色変換演算を実行するためのデータは、前記タグの階層の第四のレベルにある第四のタグを指す第三のポインタデータを含むことを特徴とする請求項16に記載されたカラー画像処理装置。

【請求項18】 前記メモリは、前記第四のタグにテーブルデータを格納し、

前記処理手段は、前記標準化された変換の代わりに、ある色空間から別の色空間へ前記入力画像データを変換するために、前記第四のタグに格納されたテーブルデータに基づき、さらに、前記第三のタグにより定義される色変換演算を実行することを特徴とする請求項17に記載されたカラー画像処理装置。

【請求項19】 前記判定手段は、前記プロファイルフォーマットに格納されたヘッダからCMMデータを読取り、そのCMMデータに対応するデータを前記第一のタグで探索することにより、前記第一のタグにアクセスすべきか否かを判定することを特徴とする請求項16に記載されたカラー画像処理装置。

【請求項20】 前記処理手段は、カラーマッチングソフトウェアを含み、ユーザの入力に従い前記カラーマッチングソフトウェアを実行することを特徴とする請求項16に記載されたカラー画像処理装置。

【請求項21】 前記処理手段は、前記読取手段により前記第三のタグから読取られたデータと、予め格納された色変換演算のリストとを整合することで、実行する色変換演算を判定することを特徴とする請求項16に記載されたカラー画像処理装置。

【請求項22】 ある色空間から別の色空間への標準化された変換に使用される数値情報が格納され、タグが階層格納されているプロファイルフォーマットにおいて、前記タグを使用して前記標準化された変換を変更するカラー画像処理方法であって、

カラー画像データを入力する入力ステップと、タグの階層の最高位にある第一のタグの所定の位置に、前記タグの階層の第二の高位にある第二のタグ中の色変換シーケンスを指す第一のポインタデータを格納する第一の格納ステップと、

前記タグの階層の第三の高位にある第三のタグに格納された色変換演算を指す第二のポインタを含む前記色変換シーケンスを前記第二のタグに格納する第二のステップと、

前記色変換演算を実行するためのデータを第三のタグに格納する第三の格納ステップと、前記プロファイルフォーマットに格納されている情報に基づき、前記第一のタ

グにアクセスすべきか否かを判定する判定ステップと、前記第一のタグにアクセスすべき場合、前記第一のタグの前記第一のポインタの情報を読取り、前記色変換シーケンスにアクセスすべきか否かを判定する第一の読取ステップと、

前記色変換シーケンスにアクセスすべき場合は、前記色変換演算にアクセスすべきか否かを判定するために、前記色変換シーケンス中の前記第二のポインタを読取る第二の読取ステップと、

前記色変換演算にアクセスすべき場合には、前記第三のタグから色変換演算を実行するためのデータを読取る第三の読取ステップと、

前記標準化された変換の代わりに、ある色空間から別の色空間へ前記入力画像データを変換するために、前記第三のタグからのデータにより読取ったデータに基づき、前記色変換演算を行う実行ステップとを有することを特徴とするカラー画像処理方法。

【請求項23】 前記色変換演算が所定の色変換演算を含む場合、前記色変換演算を実行するためのデータは、前記タグの階層の第四のレベルにある第四のタグを指す第三のポインタデータを含むことを特徴とする請求項22に記載されたカラー画像処理方法。

【請求項24】 さらに、前記第四のタグにテーブルデータを格納する第四の格納ステップと、

前記標準化された変換の代わりに、ある色空間から別の色空間へ前記入力画像データを変換するために、前記第四のタグに格納されたテーブルデータに基づき、前記第三のタグにより定義される色変換演算を実行する第二の実行ステップとを有することを特徴とする請求項23に記載されたカラー画像処理方法。

【請求項25】 前記判定ステップは、前記プロフィールフォーマットに格納されたヘッダからCMMデータを読取り、そのCMMデータに対応するデータを前記第一のタグで探索することにより、前記第一のタグにアクセスすべきか否かを判定することを特徴とする請求項22に記載されたカラー画像処理方法。

【請求項26】 前記実行ステップは、前記第三の読取ステップで読取られたデータと、予め格納された色変換演算のリストとを整合することで、実行する色変換演算を判定することを特徴とする請求項22に記載されたカラー画像処理方法。

【請求項27】 ある色空間から別の色空間への標準化された変換に使用される数値情報が格納され、タグが階層格納されているプロフィールフォーマットにおいて、コンピュータが使用可能な記憶媒体に記憶され、前記タグを使用して前記標準化された変換を変更するコンピュータが実行可能な処理ステップであって、カラー画像データを入力する入力ステップと、タグの階層の最高位にある第一のタグの所定の位置に、前記タグの階層の第二の高位にある第二のタグ中の色変

換シーケンスを指す第一のポインタデータを格納する第一の格納ステップと、

前記タグの階層の第三の高位にある第三のタグに格納された色変換演算を指す第二のポインタを含む前記色変換シーケンスを前記第二のタグに格納する第二のステップと、

前記色変換演算を実行するためのデータを第三のタグに格納する第三の格納ステップと、前記プロフィールフォーマットに格納されている情報に基づき、前記第一のタグにアクセスすべきか否かを判定する判定ステップと、前記第一のタグにアクセスすべき場合、前記第一のタグの前記第一のポインタの情報を読取り、前記色変換シーケンスにアクセスすべきか否かを判定する第一の読取ステップと、

前記色変換シーケンスにアクセスすべき場合は、前記色変換演算にアクセスすべきか否かを判定するために、前記色変換シーケンス中の前記第二のポインタを読取る第二の読取ステップと、

前記色変換演算にアクセスすべき場合には、前記第三のタグから色変換演算を実行するためのデータを読取る第三の読取ステップと、

前記標準化された変換の代わりに、ある色空間から別の色空間へ前記入力画像データを変換するために、前記第三のタグからのデータにより読取ったデータに基づき、前記色変換演算を行う実行ステップとを有することを特徴とする処理ステップ。

【請求項28】 前記色変換演算が所定の色変換演算を含む場合、前記色変換演算を実行するためのデータは、前記タグの階層の第四のレベルにある第四のタグを指す第三のポインタデータを含むことを特徴とする請求項27に記載された処理ステップ。

【請求項29】 さらに、前記第四のタグにテーブルデータを格納する第四の格納ステップと、

前記標準化された変換の代わりに、ある色空間から別の色空間へ前記入力画像データを変換するために、前記第四のタグに格納されたテーブルデータに基づき、前記第三のタグにより定義される色変換演算を実行する第二の実行ステップとを有することを特徴とする請求項28に記載された処理ステップ。

【請求項30】 前記判定ステップは、前記プロフィールフォーマットに格納されたヘッダからCMMデータを読取り、そのCMMデータに対応するデータを前記第一のタグで探索することにより、前記第一のタグにアクセスすべきか否かを判定することを特徴とする請求項27に記載された処理ステップ。

【請求項31】 前記実行ステップは、前記第三の読取ステップで読取られたデータと、予め格納された色変換演算のリストとを整合することで、実行する色変換演算を判定することを特徴とする請求項27に記載された処理ステップ。

【請求項 3 2】 変更可能なタグおよび複数の予め定義された色変換シーケンスを有する所定のプロファイルフォーマットを変更し、変更されたプロファイルフォーマットに基づき、カラー画像データに色変換演算を施すカラー画像処理装置であって、前記カラー画像データ、オーバーライド情報、並びに、複数の色変換演算を含む色変換シーケンスおよび色変換演算が組み込まれたタグデータを入力する入力手段と、階層記憶構造の中で前記色変換シーケンスが前記色変換演算より高位にあり、各色変換シーケンスは色変換演算を指す少なくとも一つのポイントを含むように、前記階層記憶構造に従い、前記タグデータを前記プロファイルフォーマットの変更可能なタグに格納する格納手段と、前記オーバーライド情報が所定の値を有する場合、前記変更可能なタグに格納されたタグデータに基づき、前記入力カラー画像データを処理し、(1)前記オーバーライド情報により定義される前記変更可能なタグの中の色変換シーケンスにアクセスし、(2)アクセスされた色変換シーケンス中のポイントにより定義される色変換演算にアクセスし、(3)アクセスされた色変換演算に基づき、前記カラー画像データを処理することにより、その処理を実行する処理手段とを有することを特徴とするカラー画像処理装置。

【請求項 3 3】 変更可能なタグおよび複数の予め定義された色変換シーケンスを有する所定のプロファイルフォーマットを変更し、変更されたプロファイルフォーマットに基づき、カラー画像データに色変換演算を施すカラー画像処理方法であって、前記カラー画像データ、オーバーライド情報、並びに、複数の色変換演算を含む色変換シーケンスおよび色変換演算が組み込まれたタグデータを入力する入力ステップと、階層記憶構造の中で前記色変換シーケンスが前記色変換演算より高位にあり、各色変換シーケンスは色変換演算を指す少なくとも一つのポイントを含むように、前記階層記憶構造に従い、前記タグデータを前記プロファイルフォーマットの変更可能なタグに格納する格納ステップと、前記オーバーライド情報が所定の値を有する場合、前記変更可能なタグに格納されたタグデータに基づき、前記入力カラー画像データを処理し、(1)前記オーバーライド情報により定義される前記変更可能なタグの中の色変換シーケンスにアクセスするステップ、(2)アクセスされた色変換シーケンス中のポイントにより定義される色変換演算にアクセスするステップ、および、(3)アクセスされた色変換演算に基づき、前記カラー画像データを処理するステップを含む処理ステップとを有することを特徴とするカラー画像処理方法。

【請求項 3 4】 コンピュータが読取可能な記憶媒体に格納され、変更可能なタグおよび複数の予め定義された

色変換シーケンスを有する所定のプロファイルフォーマットを変更し、変更されたプロファイルフォーマットに基づき、カラー画像データに色変換演算を施すコンピュータが実行可能な処理ステップであって、

前記カラー画像データ、オーバーライド情報、並びに、複数の色変換演算を含む色変換シーケンスおよび色変換演算が組み込まれたタグデータを入力する入力ステップと、

階層記憶構造の中で前記色変換シーケンスが前記色変換演算より高位にあり、各色変換シーケンスは色変換演算を指す少なくとも一つのポイントを含むように、前記階層記憶構造に従い、前記タグデータを前記プロファイルフォーマットの変更可能なタグに格納する格納ステップと、

前記オーバーライド情報が所定の値を有する場合、前記変更可能なタグに格納されたタグデータに基づき、前記入力カラー画像データを処理し、(1)前記オーバーライド情報により定義される前記変更可能なタグの中の色変換シーケンスにアクセスするステップ、(2)アクセスされた色変換シーケンス中のポイントにより定義される色変換演算にアクセスするステップ、および、(3)アクセスされた色変換演算に基づき、前記カラー画像データを処理するステップを含む処理ステップとを有することを特徴とする処理ステップ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ともにプロファイルフォーマットで定義される色変換演算であって、他の色変換演算に優先するある順序の色変換演算と、プロファイルフォーマットにおける色変換演算を指示するポイントとをプライベートタグに格納することにより、International Color Consortium (ICC) プロファイルフォーマットのようなプロファイルフォーマットを変更するシステムに関するものである。

【0002】

【従来の技術】プロファイルフォーマットは、一つのデバイスの色空間に対応してフォーマットされたカラー画像データを、別のデバイスの色空間、または、デバイスから独立した色空間（以下「デバイス独立色空間」と呼ぶ）に対応してフォーマットされたカラー画像データに変換するために使用される。例えば、ICC プロファイルフォーマットは、RGB（赤、緑および青）色空間に対応してフォーマットされたカラー画像データを「プロファイル結合空間」と呼ばれるデバイス独立色空間に変換し、次に、カラープリンタのCMY（シアン、マゼンタおよびイエロー）色空間に対応してフォーマットされたカラー画像データに変換するための一連の標準色変換演算を定義する。

【0003】標準化されたICCプロファイルは、場合によっては異なるデバイスの間に色の互換性を成立させる

が、標準化されたICCプロファイルが実行する色変換演算の数と種類、および、色変換を実行する順序に関しては融通性を欠いている。

【0004】すなわち、ICCプロファイルはパブリックタグの中で色変換演算を定義する。一般に、タグは、色変換演算を実行するために使用されるフォーマットされたデータ、および/または、他のタグを指示するポインタを記憶する一つのメモリ領域である。ICCにより定義されるパブリックタグは、所定の一連の色変換を所定の順序で実行する標準化された色変換演算を容れるタグから構成されている。ICCパブリックタグの変更は不可能である。

【0005】これに関して、ICCパブリックタグはICCで定義済みの色変換演算に限定されているから、いくつかの色変換演算、とくに、ICCでは対応できない色空間相互の変換に必要とされるような色変換演算を実行することができない。

【0006】さらに、ICCパブリックタグの変更は不可能であるから、色変換を実行するためにパブリックタグを一度アクセスしたならば、そのパブリックタグに格納されているプリセットされた色変換演算のすべてを指定された順序で実行しなければならない。場合によっては、その結果、色変換に誤りを生じることもある。また、単に効率が低下するだけの場合もある。例えば、色変換演算が必要であるか否かに関わらず、入力画像データに対して不必要な色変換演算が実行されてしまうこともある。

【0007】従って、ソフトウェア開発者がICCプロファイルフォーマットに格納された色変換演算の種類、数および順序を変更することを可能にするようなICCプロファイルフォーマットを変更するシステムが必要である。

【0008】

【発明の概要】本発明は、ICCプロファイルにおける標準化された色変換演算をプライベートタグを使用して変更するシステムを提供することによって、上記の要求に対処するものである。本発明によれば、標準化された色変換演算に加えて、色変換演算を格納し、ICCプロファイルを通じてアクセスする。また、本発明には、色変換演算をアクセスするためにプライベートタグの中のポインタを使用することにより、パブリックタグに格納されている色変換演算にランダムアクセスすることができるという利点もある。

【0009】すなわち、一つの面によれば、本発明は、プロファイルフォーマットのパブリックタグに格納された所定の一連の色変換演算の代わりに、第一のタグおよび第二のタグを使用して色変換演算を実行し、入力画像データに色変換演算を行うシステム（すなわち、方法、装置およびコンピュータが実行可能なプロセスのステップ）である。システムは、カラー画像データを入力する

入力ステップと、前記所定の一連の色変換演算を無効にするオーバーライド情報を格納する前記第一のタグと、階層記憶構造を介してアクセス可能な色変換演算データを格納する前記第二のタグとを格納する格納ステップと、前記第一のタグのオーバーライド情報に基づき、前記第二のタグにアクセスすべきか否かを判定する判定ステップとを含む。システムには、前記判定ステップで前記第二のタグにアクセスすべきであると判定された場合、前記第二のタグの色変換演算データにアクセスするための前記階層記憶構造のポインタに従い、前記色変換演算データを読取る読取ステップ、および、前記読取ステップで読取られた色変換演算データに基づき、前記入力画像データに色変換演算を施す処理ステップも含まれている。

【0010】上記の構成により、標準プロファイルフォーマットにより提供される特徴の利点を維持しつつ、モニタ用のRGBカラー画像データなどのデバイスにする第一の色空間から、デバイス独立色空間へ、カスタマイズされた色変換プロセスを使用して、カラー画像データを変換することができる。上記のシステムを二つの異なるデバイスで使用すれば、データを第一のデバイスの第一の色空間からデバイス独立色空間へ変換し、次に、デバイス独立色空間から第二のデバイスの第二の色空間へ変換することが可能である。

【0011】さらに、上記のシステムは、階層構造に基づいて色変換演算を格納し呼び出すので、カスタマイズされた色変換演算を容易にアクセスするカラーマッチングソフトウェアを提供する。

【0012】別の一面によれば、本発明は、標準化された変換により使用される数値情報が格納され、かつ、タグが階層格納されているプロファイルフォーマットにおいて、タグを使用して標準化された変換を変更するシステムである。システムは、カラー画像データを入力する入力ステップと、タグの階層の最高位にある第一のタグの所定の位置に、前記タグの階層の第二の高位にある第二のタグ中の色変換シーケンスを指す第一のポインタデータを格納する第一の格納ステップとを含む。システムには、前記タグの階層の第三の高位にある第三のタグに格納された色変換演算を指す第二のポインタを含む前記色変換シーケンスを前記第二のタグに格納する第二のステップ、および、前記色変換演算を実行するためのデータを第三のタグに格納する第三の格納ステップも含まれている。判定ステップは、前記プロファイルフォーマットに格納されている情報に基づき、前記第一のタグにアクセスすべきか否かを判定し、第一の読取ステップは、前記第一のタグにアクセスすべき場合、前記第一のタグの前記第一のポインタの情報を読取り、前記色変換シーケンスにアクセスすべきか否かを判定する。

【0013】第二の読取ステップは、前記色変換シーケンスにアクセスすべき場合、前記色変換演算にアクセスすべきか否かを判定するために、前記色変換シーケンス

中の前記第二のポイントを読取り、第三の読取ステップは、前記色変換演算にアクセスすべき場合、前記第三のタグから色変換演算を実行するためのデータを読取る。実行ステップは、前記標準化された変換の代わりに、ある色空間から別の色空間へ前記入力画像データを変換するために、前記第三のタグからのデータにより読取ったデータに基づき、前記色変換演算を行う。

【0014】本発明の上記の面によれば、ユーザは、プロフィールフォーマットにより実行される色変換演算の全体を変更するために使用できるプライベートタグを選択的に作成および変更し、その後、変更された色変換演算をカラー画像データに対して実行することができるので有利である。さらに、上記の階層記憶構造は、プライベートタグの作成および変更使用される色変換演算のデータへのアクセスを容易にする。

【0015】さらに別の一面によれば、本発明は、複数の変更可能なタグと、複数の事前に定義済みの色変換シーケンスとを有する所定のプロフィールフォーマットを変更し、かつ、変更されたプロフィールフォーマットに基づきカラー画像データに対して色変換演算を実行するシステムである。システムは、前記カラー画像データ、オーバーライド情報、並びに、複数の色変換演算を含む色変換シーケンスおよび色変換演算が組み込まれたタグデータを入力する入力ステップを含む。格納ステップは、階層記憶構造の中で前記色変換シーケンスが前記色変換演算より高位にあり、各色変換シーケンスは色変換演算を指す少なくとも一つのポイントを含むように、前記階層記憶構造に従い、前記タグデータを前記プロフィールフォーマットの変更可能なタグに格納する。処理ステップは、前記オーバーライド情報が所定の値を有する場合、前記変更可能なタグに格納されたタグデータに基づき、前記入力カラー画像データを処理する。

【0016】前記処理ステップは、(1)前記オーバーライド情報により定義される前記変更可能なタグの中の色変換シーケンスにアクセスするステップ、(2)アクセスされた色変換シーケンス中のポイントにより定義される色変換演算にアクセスするステップ、および、(3)アクセスされた色変換演算に基づき、前記カラー画像データを処理するステップを含む。

【0017】以上の要約は、本発明の本質をすばやく理解させるためのものである。添付する図面と関連させた、好ましい実施形態の詳細な説明を参照することにより、本発明をより完全に理解することができる。

【0018】

【好ましい実施形態の詳細な説明】図1は、本発明の代表的な一実施形態の外観を示す図である。図1に示すコンピュータ装置1は、MacintoshまたはMicrosoft(R) Windowsなどのウィンドウズ環境を有するIBM PCコンパチブルコンピュータのような装置である。コンピュータ装置1は、カラーモニタなどの表示スクリーン2と、テキス

トデータおよびプログラムの指令を入力するためのキーボード4と、表示スクリーン2に表示されたオブジェクトを指示しかつ操作するためのマウスなどのポインティングデバイス6と、コンピュータ装置1により生成されたカラー画像を出力するプリンタ16とを備えている。

【0019】コンピュータ装置1は、図2に示すコンピュータディスク7のような大容量記憶デバイスを含む。このコンピュータディスク7には、パブリックタグおよびプライベートタグを含むICCプロフィール8のようなプロフィールフォーマット、DOS(R)オペレーティングシステム(以下「OS」という)、並びに、Microsoft Windows(R)のようなウィンドウズOSとが格納されている。同様に、コンピュータディスク7には、キヤノン(R)カラーマッチングソフトウェア(以下「CCM」という)9、プライベートタグを作成および変更するアプリケーションプログラム10、並びに、プロフィールマネージャルーチン12が格納されている。これらはすべて、ディスク7のデータファイルの操作および格納、並びに、表示スクリーン2を介してオペレータへそれらファイル内のデータを提示するといったコンピュータ装置1のプログラム命令を格納している。これらのプログラムについては、以下で、さらに詳細に説明する。

【0020】コンピュータ装置1は、さらに、フロッピーディスク(以下「FD」という)を差し込むことができるフロッピーディスクドライブ(以下「FDD」という)のインタフェース14を含む。そのようなFDからの情報は、コンピュータディスク7にダウンロードすることができる。このような情報は、CMM(Color Management Module)9のようなデータファイルおよびアプリケーションプログラム、プライベートタグを作成および変更するアプリケーションプログラム10、並びに、プロフィールマネージャルーチン12が挙げられる。コンピュータ装置1は、さらに、CD-ROMインタフェース(図示せず)を含むことができ、このインタフェースからの情報もディスク7へダウンロードすることができる。

【0021】カラー画像データは、文書または他の画像を走査して、それらのビットマップ画像をコンピュータ装置1に提供するスキャナ15によって入力される。カラー画像データは、ネットワークインタフェース17のような様々な他の供給源から、または、ファクシミリ/MODEMインタフェース18を介した他の外部デバイスからコンピュータ装置1に入力してもよい。

【0022】なお、さらに、ネットワークインタフェース17のような様々な他の供給源の、または、ファクシミリ/MODEMインタフェース18を介した他の外部装置のICCプロフィール8が、コンピュータ装置1によりアクセスされる。

【0023】図1にはプログラム可能な汎用コンピュータの構成を示したが、本発明を実施する際は、専用のコンピュータ、スタンドアローンのコンピュータ、また

は、その他の種類のデータ処理装置を使用できることを理解すべきである。

【0024】図2は、コンピュータ装置1の内部構成の詳細を示すブロック図である。図2に示すように、コンピュータ装置1は、コンピュータバス21とインタフェースする中央処理装置（以下「CPU」という）20を含む。同様に、スキャナインタフェース22、ネットワークインタフェース17、ファックス/MODEMインタフェース18、ディスプレイインタフェース23、キーボードインタフェース25、マウスインタフェース29、メインメモリ30、ディスク7、FDDインタフェース14およびプリンタインタフェース24も、コンピュータバス21とインタフェースしている。

【0025】Microsoft Windows(R)、CMM 9、プライベートタグを作成および変更するアプリケーションプログラム10、プロファイルマネージャルーチン12およびその他のアプリケーションプログラム（図示せず）のような格納されたプログラム命令を実行する際に、CPU20が使用するランダムアクセスメモリを提供するように、メインメモリ30はコンピュータバス21とインタフェースしている。すなわち、CPU20は、それらのプログラムをディスク7またはFDDインタフェース14のFDの何れかからメインメモリ30へロードし、メインメモリ30に格納されたそれらのプログラムを讀出して実行する。

【0026】CMM 9を含むカラー画像処理システムに関して本発明を説明する。簡単に言えば、CMM 9は、RGBのような第一のカラーフォーマットのカラー画像データを入力し、そのカラー画像データをCMYのような第二のフォーマットに変換する。上記の色変換動作を実行するために、CMM 9はICCプロファイルに格納されたデータを使用する。このデータは「International Color Consortium Profile Format」第3.01版（1995年5月8日改訂）に解説されていて、その内容は参考として本出願にも取り入れられている。

【0027】要するに、ICCプロファイルは、あるデバイスで作成されたカラーデータを別のデバイスに固有の色空間に変換する、という対ごとに使用することができるデバイスプロファイルである。例えば、先に説明したとおり、ICCプロファイルを使用して、モニタ用のRGBカラー画像データを、プリンタ用のCMY画像データに変換することができる。

【0028】ICCプロファイルは、特定のデバイスに関する色変換情報をCMM 9に提供する。さらに詳細に言えば、ICCプロファイルは、デバイスごとに提供され、カラー画像データをデバイスに從属する色空間（以下「デバイス從属色空間」という）からプロファイル結合空間へ変換し、さらにプロファイル結合空間から別のデバイス從属色空間へ変換する際に、CMM 9により使用される。この関係を図3に示す。

【0029】すなわち、図3は、モニタ31のRGB画像デー

タをプロファイル結合空間34のデバイスから独立した画像データに変換する際に使用されるICCプロファイル32を有するモニタ31を示す。プリンタ35は、デバイスから独立したプロファイル結合空間のカラー画像データを、プリンタ35が利用可能なCMY画像データに変換するICCプロファイル37を含む。このように、CMM 9は、二つのデバイス從属色空間の間で変換を行うためにICCプロファイルを使用する。なお、図3は、モニタ31とプリンタ35に関してそれぞれICCプロファイル32および37を示しているが、それらのICCプロファイルがそれぞれ対応する装置に常駐しているわけではない。それらのICCプロファイルは、ICCプロファイルに従って変換されるべきデータに埋め込まれてもよく、または、接続されたパーソナルコンピュータのメモリに記憶されていてもよい。例えば、ICCプロファイルは、単一のCPUによりアクセス可能な単一のメモリに格納させることが可能であろう。

【0030】さらに、ICCプロファイルは、プリンタやモニタのほかにも、スキャナ、ファクシミリ装置、その他のデバイスにも使用することができる。

【0031】プロファイル結合空間は、D50標準光源と、1931 CIE標準オブザーバと、0/45または45/0反射率測定ジオメトリとにより規定される。基準ビューイング条件は、D50アーツビューイング環境であるANSI PH2.30-1989である。

【0032】図4に一例を示すICCプロファイルは、ヘッダ39およびタグテーブル40という二つの基本要素を含む。ヘッダ39は、ICCプロファイルに従って入力画像データを処理するために、CMM 9により使用される情報を含む。ヘッダデータはビッグエンジアン(big-endian)表記でなければならない。以下で、図4を参照してさらに詳細に説明するタグテーブル40は、パブリックタグおよびプライベートタグを介して色変換演算およびその他の情報をアクセスするために、CMM 9により使用される。

【0033】ICCプロファイルには、CMMがプロファイル結合空間とデバイス從属色空間との間でカラー情報を変換するために必要な一連の情報を不足なく提供するように設計されたリカードパブリックタグ(required public tag)が含まれている。さらに、ICCプロファイルは、付加変換を行うために使用することができるオプションパブリックタグ(optional public tag)、および、個々の開発者がそのICCプロファイルに専用の値を追加するためにカスタマイズできるプライベートタグを含んでいてもよい。

【0034】色変換を行うために、CMM 9は、スキャナなどの入力デバイスに対して、プロファイルが下記のタグを有することを要求する。つまり、プロファイル記述タグ、デバイスメーカータグ、デバイスモデル名タグ、メディアXYZ白色点タグ(mediaXYZ white point tag)、UCC MSプライベート情報タグ、著作権タグ、赤チャネルの相対XYZ三刺激値を含む赤の色剤タグ(red colorant ta

g)、青チャネルの相対XYZ三刺激値を含む青の色剤タグ(blue colorant tag)、緑チャネルの相対XYZ三刺激値を含む緑の色剤タグ(red colorant tag)、赤チャネルの階調再現曲線タグ、緑チャネルの階調再現曲線タグおよび青チャネルの階調再現曲線タグである。オプションとして、プロファイルは8ビットまたは16ビットのLUTを定義するAtoBnタグを含むことができる。

【0035】色変換を行うために、CMM 9は、モニタなどの表示デバイスに対して、プロファイルが下記のタグを有することを要求する。つまり、プロファイル記述タグ、デバイスメーカータグ、デバイスモデル名タグ、メディアXYZ白色点タグ、著作権タグ、赤の蛍光体の相対値を含む赤の色剤タグ、青の蛍光体の相対値を含む青の色剤タグ、緑の蛍光体の相対値を含む青の色剤タグ、赤チャネル階調再現曲線タグ、緑チャネル階調再現曲線タグおよび青チャネル階調再現曲線タグである。オプションとして、プロファイルはUCCMSプライベート情報タグを含むことができる。

【0036】色変換を行うために、CMM 9は、プリンタなどの出力デバイスに対して、プロファイルが下記のタグを有することを要求する。つまり、プロファイル記述タグ、デバイスメーカータグ、デバイスモデル名タグ、AtoB0タグ、BtoA0タグ、ガミュータタグ(gamut tag)、AtoB1タグ、BtoA1タグ、AtoB2タグ、BtoA2タグ、UCCMSプライベート情報タグ、XYZメディア白色点タグ(XYZ media white point tag)、測定タグおよび著作権タグである。

【0037】上記のAtoBnタグはICC lut8Type またはICC lut16Type 構造の何れかを有する。ICC lut8Type またはICC lut16Type 構造の汎用モデルは次のとおりである。

マトリクス →; 一次元LUT →; 多次元LUT →; 一次元LUT

【0038】lut8Type タグの場合、入力LUT、出力LUTおよびカラーLUTは、8ビット符号なし値のアレイである。各入力テーブルは1バイト整数から構成される。また、入力テーブルのエントリはそれぞれ、0から255の範囲に適切に標準化されている。タグに格納される場合、一次元LUTは、ICC仕様に基づき、昇順に順次パックされるものと想定される。

【0039】AtoB0タグは、写真レンダリングに使用される。AtoB0タグは、3×3マトリクスを定義し、そのマトリクス要素はAtoB0タグのバイト12から45に格納され、C, M, Y入力の入力チャネルはバイト8に格納され、L, a, b出力の出力チャネルはバイト9に格納され、LUTグリッドポイント(例えば、33×33×33)はバイト10に格納され、パディング(padding)はバイト11に格納され、入力テーブル(一致)はバイト46以降に格納され、カラーLUTおよび出力テーブルも写真レンダリング用として格納されている。バイト0から3はタグを定義する。

【0040】BtoA0タグは、入力テーブルと出力テーブ

ルが入れ替わり、カラーLUTがAtoB0タグの逆であることを除いて、AtoB0タグと同じフォーマットを有する。

【0041】ガミュータタグは、ガミュータタグが入力チャネルに対してL, a, b値を使用し、出力チャネルにビットマップを出力することを除いて、AtoB0タグと同じフォーマットを有する。

【0042】AtoB1タグは、色差最小レンダリング(relative colormetric rendering)に使用され、AtoB0タグに対応するフォーマットを有する。一方、BtoA1タグは、色差最小レンダリングに使用され、BtoA0タグに対応するフォーマットを有する。これらはともに lut16Type または lut8Type の何れかである。

【0043】AtoB2タグは、彩度優先レンダリング(saturation rendering)に使用され、AtoB0タグに対応するフォーマットを有する。一方、BtoA2タグは、彩度優先レンダリングに使用され、BtoA0タグに対応するフォーマットを有する。これらはともにlut16Typeまたはlut8Typeの何れかである。

【0044】CMM 9は、同様に、オプションのプレビュープロファイルのパブリックタグ(optional preview profile public tag)を支援する。このタグは、入力チャネルにL, a, bデータが入力されることを除いて、AtoB0タグと同一のフォーマットを有する。

【0045】以下で、さらに詳細に説明するが、本発明は、カスタマイズされたプライベートタグを使用して、所定のICCプロファイル内で実行される色変換演算を変更および操作する。

【0046】本発明において、CMM 9は、ICCプロファイル8、および/または、ICCプロファイル8に格納されている様々なパブリックタグおよびプライベートタグにアクセスして、プライベートタグ(および/またはデバイスプロファイル)を作成または変更するプロファイルマネージャルーチン12(例えば、Canon Information Systems, Inc. が製造している ColorGear(R))を含む。

【0047】プライベートタグの作成または変更は、図5に示す、プライベートタグに情報を入力するためのプライベートタグ作成および変更アプリケーションプログラム10の動作を記述する流れ図に従って実行される。さらに詳しくは、このアプリケーションプログラム10を実行する場合、ステップS501で、プライベートタグを作成すべきか、変更すべきかを判定する。プライベートタグを作成すべきであれば、処理はステップS502へ進み、表示スクリーン2に表示することで、プライベートタグを作成するために必要な情報をユーザに催促する。ユーザが必要なすべての情報を入力しきれなかった場合は、プライベートタグは作成されない。ユーザに催促する画面の例を図6および図7に示す。図6および図7に示す画面43および44はそれぞれ、以下で、さらに詳しく説明する図8のプライベートタグに対応している。

【0048】これに対し、ステップS501で、プライベート

トタグを変更すべきである場合、処理はステップS506へ進み、ユーザは入力画面43（または入力画面44）の「Get」ボタン42をクリックする。これにตอบสนองしてアプリケーションプログラム10は、画面の情報に基づき、プロファイルマネージャーテン12にプライベートタグ情報を要求する。プロファイルマネージャーテン12は、要求されたプライベートタグデータのメモリ領域をアクセスし、要求されたデータをアプリケーションプログラム10に供給する。

【0049】プライベートタグデータを受信すると、ステップS502において、アプリケーションプログラム10はユーザに対して情報を画面43および44に表示する。そこで、ユーザは、画面43および/または44で情報を入力し、「Set」ボタン46をクリックすることにより、希望のどおりにプライベートタグに情報を設定することができる。ユーザは、「Next」ボタン47および「Previous」ボタン48を使用して、図6および図7に示されるプライベートタグデータを見通すことができる。

【0050】ステップS503で情報が入力されると、処理はステップS504へ進み、ユーザは「Set」ボタン46をクリックすることによりプライベートタグデータを設定する。ユーザがプライベートタグデータを設定すると、プライベートタグデータはアプリケーションプログラム10からプロファイルマネージャーテン12へ送られる。プライベートタグデータを受信すると、ステップS505で、プロファイルマネージャーテン12は、プライベートタグの中で規定されるバイト割当てに従い、プライベートタグデータを記憶する。例えば、図8に示し、以下で、さらに詳しく説明するucmIプライベートタグの場合、バイト4から7に「キャノンシグネチャ(Canon Signature)」を格納し、バイト112から175には「作成部門(Creator Division)」を格納する、などになっている。このように、ユーザはプライベートタグの中に格納されている情報を変更することができる。

【0051】先に説明したように、図4は、ヘッダ39およびタグテーブル40を含むICCプロファイル8の一例について、その画面表示のプリントアウトを示す。

【0052】ヘッダ39は、ICCプロファイル8に特有の一群のパラメータを提供し、プロファイルの先頭の128バイトに格納されているのが好ましい。ヘッダ39に含まれるパラメータを以下に記述する。

【0053】(1)サイズ(Size): ヘッダ39のバイト0から3に格納され、プロファイルのサイズを定義する。

【0054】(2)CMMタイプ(CMMType): ヘッダ39のバイト4から7に格納され、プロファイルが関連しているCMMを定義する。Canon(R)のデバイスの場合、この値は「UCM」である。

【0055】(3)バージョン(Version): バイト8から11に格納され、プロファイルのバージョン番号を定義する。これはICCにより2000000Hからと定義されている。

【0056】(4)プロファイルクラス(ProfileClass): バイト12から15に格納され、プロファイルのクラスを定義し、「prtr」(プリンタ)、「mntr」(モニタ)、「scnr」(スキャナ)、「link」(リンクデバイス)、「spac」(色空間変換)および「abst」(アブストラクトプロファイル)の何れか一つであればよい。

【0057】(5)データ色空間(DataColorSpace): バイト16から19に格納され、プロファイルがカラー画像データを変換するときのカラーフォーマットを定義し、RGB、XYZ、GRAY(グレースケール)、CMY、Luv、HSV、CMYK、YCbR、HLS、LabおよびYxyの中の何れか一つであればよい。

【0058】(6)インタチェンジ空間(InterchangeSpace): バイト20から23に格納され、プロファイル結合空間を定義し、LabまたはXYZの何れか一つであればよい。

【0059】(7)作成日(CreationDate): バイト24から35に格納され、プロファイルが作成された日付および時間を定義する。

【0060】(8)CS2シグネチャ(CS2Signature): バイト36から39に格納され、プロファイルのファイルシグネチャを定義する。ファイルシグネチャは、そのプロファイルを使用するデバイスのオペレーティングシステムにより、アイコンを作成するために使用される。

【0061】(9)一次プラットフォーム(Prim. platform): バイト40から43に格納され、プロファイルが作成された一次プラットフォームまたはOSを定義し、その値は、「Appl」(Apple OS)、「MSFT」(MicrosoftのOS)、「SGI」(Silicon Graphics)、「SUNW」(Sun)および「TGN」(Taligent)の何れか一つであればよい。

【0062】(10)フラグ(Flags): バイト44から47に格納され、分散処理およびキャッシングオプションなどのCMMに関する様々なヒントを定義する。このパラメータは、キャノン(R)のデバイスでは使用されないため、0Hに設定される。

【0063】(11)デバイス製造者(deviceManufacturer): バイト48から51に格納され、プロファイルが使用されるべきデバイスの製造者のシグネチャである。キャノン(R)のデバイスの場合、このパラメータは値「CANO」を有する。

【0064】(12)デバイスモデル(deviceModel): バイト52から55に格納され、プロファイルが使用されるべきデバイスのモデル番号または名を定義する。「デバイスモデル」の値は、ICC、AppleのColorSync(R)およびMicrosoftのICM(R)(Image ColorMatching)により課されている規格に準拠していなければならない。とくに、モデル番号または名は、「A」から「Z」までの文字(大文字のみ、および、「0」から「9」の文字)を使用する4バイトのASCII文字列でなければならない。「デバイスモデル」のフォーマットは次のようにすべきである。

1バイト目: 部門ID(Division ID)

2-4バイト目: モデル番号および拡張(extension)

【0065】「部門ID」は、その製品を製造した会社または部門を識別する。理想的には、それぞれの会社または部門が独自の「部門ID」を有するべきである。キヤノンにおいては、部門は次のように定義されている。

「B」はバブルジェットを、「C」は複写機を、「D」はデジタルカメラを、「F」はファクシミリを、「L」はレーザビームプリンタを、「M」はCanon(R)のモニターを、「S」はスキャナを、「V」はビデオカメラをそれぞれ表し、「Z」はサードパーティ、つまりキヤノン以外の製品ファイルを表す。これらの値は無作為に定められ、希望に応じて設定できる。好ましい実施形態においては、「デバイスモデル」のバイト2から4は、下記の二つのフォーマットのうち一方に従って符号化される。

【0066】(a)フォーマット1

1バイト: 部門バイト

2バイト: モデル名

1バイト: 拡張(extension)

【0067】前述したように、部門バイトは、プロファイルを使用するデバイスの部門を格納する。モデル名バイトは、デバイスのモデルを格納する。拡張バイトはデバイスの拡張を格納し、例えば、BJ-600efはBJ-600デバイスの拡張であると考えられるであろう。二つのデバイスが同じ「デバイスモデル」を有する事態を避けるため、次の規約を採用している。A-Iは九つの拡張を表す拡張標識として使用され、J-Rは、先頭の3バイトに衝突がある場合に、次の九つの拡張を表す標識として使用され、S-Zは次の八つの拡張を表す拡張標識として使用され、0-9は次の10の拡張を表す拡張標識として使用される。このシステムの一例を以下に示す。

BJC-4000 →; B40A (BJC-4000の第一の拡張)、
BJC-4000E →; B40B (BJC-4000の第二の拡張)、
BJC-400 →; B40J (BJC-400の第一の拡張)、および
BJC-400X →; B40K (BJC-400の第二の拡張)

【0068】(b)フォーマット2

1バイト: 部門バイト

3バイト: モデルID + 拡張

【0069】このフォーマットでは、部門バイトは先に説明した部門バイトと同じであり、モデルIDおよび拡張番号は、二つの異なるデバイスが同じ「デバイスモデル」をもたないことを保証するように定められている。第二のフォーマットにおいては、数学アルゴリズムを使用して、実際のモデル番号、例えばBJC-600の「600」から独自のモデルID+拡張を計算する。ただし、デバイスごとに独自の「デバイスモデル」値を生成するのであれば、どのような方法を使用してもよい。

【0070】(13)デバイス属性(deviceAttributes): バイト56から63に格納され、特定のデバイスセットアップに独自の属性である。本発明においては、媒体タイプ、解像度、ハーフトーンを生成するインクタイプおよびク

リエータ(halftoning ink type and creator)を指定しなければならない。「デバイス属性」の好ましいフォーマットは、バイト56から57をICC予約とし、バイト58を媒体タイプにし、バイト59を解像度にし、バイト60を階調処理(screening)にし、バイト61をインクタイプにし、バイト62から63を将来の用途に予約したものである。

【0071】好ましい実施形態においては、媒体タイプ、解像度、階調処理およびインクタイプは次のように定義された値を有する。

(a)媒体タイプ

0: 適用不可/ドントケア

1: 普通紙

2: コート紙

3: 光沢紙

4: OHP記録紙

5: 強光沢フィルム(High Glossy Film)

6: ファインコート紙

7: BPF記録紙

8: 織物記録紙(Textile Paper)

(b)解像度

0: 適用不可/ドントケア

1: 1インチ当り180×180ドット(180×180dpi)

2: 200×200dpi

3: 300×300dpi

4: 360×360dpi

5: 400×400dpi

6: 600×300dpi

7: 600×600dpi

8: 720×360dpi

9: 720×720dpi

10: 1200×600dpi

11: 1200×1200dpi

(c)階調処理

0: 適用不可/ドントケア

1: パターン1

2: パターン2

3: パターン3

4: 誤差拡散

5: 連続階調パターン1

6: 連続階調パターン2

7: 連続階調パターン3

(d)インクタイプ

0: 適用不可/ドントケア

1: 普通インク

2: インクタイプ1

【0072】希望に従い、上記の値を変更することができる。

【0073】(14)レンダリング特性(RenderingIntent): バイト64から67に格納され、プロファイル設計の趣旨

(intent) (すなわち、色味重視(perceptual)、忠実再現(relative colormetric)、彩度重視(saturation)および絶対(absolute))を定義する。本発明において、この値は0である。

【0074】(15)白色XYZ(WhiteXYZ): バイト68から79に格納され、プロファイル結合空間の照度値を定義する。本発明において、この値はD50に設定される。

【0075】(16)バイトは80から127は将来の使用に備えてICCにより予約されている。

【0076】ヘッダ39は、プロファイルヘッダに格納できるすべての情報を限なく含むリストではなく、単に、プロファイルヘッダに格納できる情報の一例を示しているのに過ぎない。

【0077】以上説明した情報に加えて、必要に応じてまたは希望に従い、他の情報をヘッダに追加してもよい。この目的のために、先に示したように、将来の使用に備えて、ICCは48バイトを予約している。しかしながら、本発明を実現するためには、ヘッダ39に示すすべての情報が存在していなければならない。

【0078】タグテーブル40は、パブリックおよびプライベートともに、タグのリストと、それらのタグに関する情報とを含む。プロファイルの中で、タグテーブル40はバイト128 (すなわち、ヘッダ39の後) から始まる。より詳しくは、タグテーブル40に示すように、「Ind」はタグテーブル40中のタグ番号を示し、「Signat」はタグシグネチャであって、CMMにより特定のタグの位置を確定するために使用され、「エレメントオフセット(elementOffset)」はタグが始まるメモリの位置を10進数と16進数の両方で示し、「サイズ(size)」はバイト単位でタグのサイズを定義する。

【0079】タグテーブル40中の個々のタグは、そのバイト0から3がタグシグネチャを定義し、バイト4から7がタグデータの開始を指示するオフセット値を定義し、バイト8から11はタグのバイト数を定義するような構造である。

【0080】先に述べたように、パブリックタグは、すべてのICCプロファイルで使用することができる普通の(stock)色変換演算を定義する。タグテーブル40で規定されるパブリックタグの一例は、3×3色変換マトリクス処理、三次元LUT、および、三つの一次元LUTからなる二組を含むA2B2である。パブリックタグの他の例は、ICCプロファイルフォーマット文書にさらに詳細に記述されている。

【0081】タグテーブル40において、0から16の「Ind」値を有するタグはパブリックタグを構成し、17と18の「Ind」値を有するタグはプライベートタグを構成する。この場合、プライベートタグは、キャノン(R)登録のプライベートタグである。

【0082】本発明は、タグテーブル40にリストされているパブリックタグと関連させながら、上記の二つのプ

ライベートタグに関連して説明される。ただし、本発明は、ICCプロファイルフォーマット文書に記載されたもののパブリックタグ、または、それにコンパチブルなどのようなタグと組合わせても使用することができる。

【0083】要するに、本発明は、入力画像データに色変換演算を施すシステムであり、このシステムは、プロファイルフォーマット中の一つのパブリックタグに格納された所定の一連の色変換演算を行う代わりに、第一のタグおよび第二のタグを使用して色変換演算を行う。このシステムは次のステップを含む。つまり、カラー画像データを入力する入力ステップと、所定の一連の色変換を無効にする(override)オーバーライド情報を格納する第一のタグ、および、階層記憶構造を介してアクセス可能な色変換演算データを格納する第二のタグを格納する格納ステップと、第一のタグのオーバーライド情報に基づき、第二のタグにアクセスするか否かを判定する判定ステップとである。このシステムには、さらに、次のステップが含まれている。つまり、判定ステップで第二のタグにアクセスすると判定した場合、第二のタグの階層記憶構造中の色変換演算データにアクセスするためのポインタに従い、第二のタグの色変換演算データを読取る読取ステップと、読取ステップで読取った色変換演算データに従い、入力画像データに色変換演算を施す処理ステップとである。

【0084】本発明は、色変換演算を格納し、階層構造中の他のパブリックおよびプライベートタグを呼出す。この階層構造を図9に示し、図に示す各要素の詳細な説明と関連させながら、図9を詳細に論じる。

【0085】以下、そのシグネチャ「ucmI」および符号52によって示す値17の「Ind」を有するプライベートタグは、ICCプロファイル8のプライベートタグとICCプロファイル8のパブリックタグとの相互作用に関する情報を記憶している。さらに、ucmI 52は、プライベートタグを使用してカラーマッチングを行うCMM 9が要求する情報を記憶している。ucmI 52が欠落している場合、CMM 9はデフォルト設定を使用して、プロファイルを解釈する。図8に示すように、この情報はバイト順にucmI 52に格納される。

【0086】すなわち、ucmI 52の列53は、列56にデータを格納しているucmI 52のバイトを定義する。列54は列53のバイトに格納されたデータを記述し、列57は列53のバイトに記憶されたデータのフォーマットを定義する。なお、ここで、「uInt32」は32ビット符号なし整数を表し、「0x」の後に数字が続く表記はその数が16進数フォーマットであることを示す。

【0087】ucmI 52には次のデータを格納する。つまり、ucmI 52にアクセスする際にCMM9が使用できるタグ特有の識別子である「uccmsプライベート情報タグシグネチャ(uccmsPrivateInformationTag Signature)」と、プライベートタグをキャノン(R)のプライベートタグと

識別するための二次識別子である「キャノンシグネチャ (Canon Signature)」と、色変換演算で使用するパラメータ (以下にさらに詳しく説明する) のパラメータデータサイズを定義する「パラメータバイトサイズ (Size of parameters in bytes)」と、プロファイルの読取りに要求される最小CMMバージョンを定義する「エンジンバージョン (Engine version)」と、プロファイルのバージョンを定義する「プロファイルフォーマット文書バージョン (ProfileFormat Document version)」と、その作成者によって割当てられたプロファイルバージョンを定義する「プロファイルバージョン (Profile version)」と、該当するプロファイルのビルド番号を定義する「プロファイルビルド番号 (ProfileBuild number)」と、プロファイルにより使用されるべき線形補間の種類を定義する「補間フラグ (Interpolation flag)」と、バイト32から71に格納され (以下でさらに詳しく説明する)、A2B2のような特定のパブリックタグを無効にする (override) ために要求されるシーケンスキャノンID番号 (Sequence Canon ID number) を格納するオーバーライド要素と、バイト72から111に格納され、例えばドラフトモードのようなプリンティングモードタイプを示すCMM 9の最適化フラグを格納する最適化フラグ値と、プロファイルを作成したキャノン (R) の部門を定義する「作成部門 (Creator Division)」と、プロファイルをサポートするキャノン (R) の部門を定義する「サポート部門 (Support Division)」と、フォンクライス (VonKries) 色変換を使用するか否かを判定するVon Kriesフラグと、予約バイトとである。

【0088】図10Aおよび図10Bは流れ図であり、CMM 9によるICCプロファイル8、並びに、それと関連するパブリックおよびプライベートタグのナビゲート処理を示している。

【0089】詳細に説明すると、ステップS1001で、CMM 9はヘッダ39にあるプロファイル8のCMMタイプ (CMMType) を読取る。CMMタイプは、値が格納されているメモリの領域を示すポインタに相当する。

【0090】次に、図10AのステップS1002で、ヘッダ39のCMMタイプに基づき、CMM 9がプロファイル8は「UCCM」プロファイルであると判定した場合、処理はステップS1003へ進み、CMM 9は、パブリックタグではなくucmI 52に応じた色変換演算を行う。すなわち、図9に示すように、ucmI 52は記憶階層の最上位にある。

【0091】すなわち、CMM 9は、ucmI 52のバイト0から3に基づいてucmI 52のシグネチャを読取り、ucmI 52に対応するメモリの「エレメントオフセット (elementOffset)」領域、この場合、バイト25667810をアクセスする (図4を参照)。

【0092】その後、図10AのステップS1003で、CMM 9は、ucmI 52のバイト32から71のデータを読取り、それに基づき、プロファイル8のパブリックタグにより定義

される演算の何れが無効にされる (override) かを判定する。ここで、ucmI 52のオーバーライド領域に0Hがあれば、CMM 9は対応するパブリックタグに格納されたプロセスに従って色変換を実行する。ucmI 52のオーバーライド領域に非ゼロ値がある場合、CMM 9は、ucmI 52に格納され、かつ、その非ゼロ値によって定義されるプロセスに従い、色変換を実行する。例えば、図8に示すように、AtoB0Tagオーバーライド59は値「0x0」または0Hを有する。この情報に基づき、CMM 9は、AtoB0Tagパブリックタグに格納されている色変換プロセスに応じて、AtoB0Tag色変換処理を行う。従って、AtoB0Tag色変換処理の流れは図10Bの終わりに向かって進む。

【0093】ところが、図8にも示されているように、BtoA2Tagオーバーライド60は、メモリの一つの領域を指示するポインタとして働く値「0x33」または33Hを有する。従って、BtoA2Tagの色変換演算について、処理は、BtoA2Tagパブリックタグではなく、ucmI 52に格納された情報に従って進む。

【0094】さらに詳しくは、図10Aに示すように、処理はステップS1005へ進み、以下ではそのシグネチャ「ucmP」と符号62によって示される「Ind」値18を有するプライベートタグが、CMMにより自動的に読取られる。

【0095】図11はucmP 62の一例を示す。図11に示すように、ucmP 62は、ucmI 52と同じフォーマットで、バイト順に情報を格納している。従って、簡潔にするため、この記憶フォーマットの詳細な説明を省略する。

【0096】ucmP 62は次の情報を格納する。つまり、CMM 9がucmP 62をアクセスするときに使用するタグ特定の識別子である「uccmsプライベートシーケンスタグテーブルシグネチャ (uccmsPrivateSequencesTagTable Signature)」と、ucmI 52に関して上述した「キャノンシグネチャ」と同一である「キャノンシグネチャ」と、ucmP 62を介して実行されるべき色変換シーケンスの数を定義する「シーケンス数 (number of Sequences)」と、特定の色変換シーケンスで実行されるべき一連の色変換演算を定義する「シーケンス構造体 (Sequences structures)」と、色変換シーケンスを実行するCMM 9により要求されるその他の関連情報とである。なお、図11は四つのシーケンス構造体しか示していないが、任意数のシーケンス構造体をucmP 62に加えることができ、シーケンス構造体の数は本発明を実行するハードウェアによってのみ制限される。

【0097】色変換シーケンスのシーケンス構造体を図12に示す。なお、色変換シーケンスにおける特定の演算は、それに含まれる他のパラメータとともに変化することがあるが、そのシーケンス構造体は固定されている。

【0098】すなわち、すべての色変換シーケンスは、CMM9が特定のシーケンス構造体をアクセスする際に使用するシーケンスである「CanonID (キャノンID)」を含む。この場合、「キャノンID」は、図8のucmI 52のBtoA

2Tagオーバーライド60に相当する33Hである。この対応によって、CMM9は、ucmI 52に基づき、ucmP 62におけるどのカラーマッチングシーケンスにアクセスするかを決定することができる。

【0099】色変換シーケンス構造体は、さらに次を含む。つまり、00Hの値を維持する「reserved bytes (予約バイト)」と、以下でさらに詳細に論じるように、色変換シーケンスにおける演算の数を定義する「numOps」と、該当するプロファイルがデータを変換するプロファイル結合空間を定義する「seqPCS」と、色変換シーケンスにより実行されるべき演算で使用されるパラメータのバイト長を定義する「length (長さ)」と、色変換シーケンスにより実行されるべき色変換演算のメモリにおける開始点を定義する「offset (オフセット)」とである。

【0100】図12に示すように、シーケンス構造体は、それに記憶されているデータのフォーマットをも定義する。この場合、図に示すように、フォーマットは「uInt 32」であり、上述したように、32ビット符号なし整数に相当する。

【0101】シーケンス構造体は、図11に示すように、ucmP 62の列65のバイト16から39に格納されるバイトストリングを含む。すなわち、図11および図12に示す例の場合、「CanonID」は33、「reserved」は00、「numOps」は04、「seqPCS」はLAB、「length」は40、「offset」は5163である。これらの値は、シーケンス構造体1のバイト16から39にデータとして格納される。同様に、シーケンス構造体2から4にも、識別可能なシーケンス構造体を格納させることができる。しかしながら、簡潔にするため、本発明の説明を上述のシーケンス構造体についてのみ進めるものとする。従って、図9に示すように、ucmP 62は階層記憶構造の中でucmI 52の次に位置し、その後様々な利用可能シーケンス構造体66から68が続いている。

【0102】ここまでの説明を要約すると、ucmI 52のBtoA2Tagオーバーライド60には33Hが存在し、かつ、シーケンス構造体1は33HのキャノンIDを有するので、CMM 9はシーケンス構造体1を読み取ることになる。

【0103】先に説明したように、各シーケンス構造体は、「numOps」値、「length」値および「offset」値を含む。CMM 9は、これらの値を使用して、シーケンス構造体に格納された色変換演算にアクセスする。さらに詳しくは、(1)それぞれの色変換演算は「length」により定義される固定長であり、(2)「numOps」により定義される色変換演算の数がわかっていて、かつ、(3)色変換演算が順に格納されているメモリ領域もわかっているため、CMM9は、各色変換演算を分析し(parse)、シーケンス構造体に格納されている情報に基づき演算を実行することができる。このプロセスについては、以下で、さらに詳しく説明する。

【0104】図13は色変換演算の構造体を示す。すなわち、色変換演算の構造体は次のように定義される。「oper」は実行されるべき色変換演算の種類を定義し、「subID (サブID)」は、以下で、さらに詳しく説明するように、特定の色変換演算に関する副識別番号を定義し、「length (長さ)」は色変換演算におけるパラメータリストの長さを定義し、「tagFlag (タグフラグ)」は、「tagFlag」が非ゼロ値を有する場合は別のパブリックまたはプライベートタグに対する呼出し(call)を定義し、「tagFlag」がゼロ値を有する場合はパラメータリストの中に記憶されている色変換演算に対する呼出し(call)を定義し、「parm」は色変換演算で使用されるパラメータリストを定義する。

【0105】本発明を使用して、現在実行できる色変換演算は七種類ある。それら七つの色変換演算には、括弧中のUCCMS演算番号によりそれぞれ定義される次が含まれる。つまり、(1)N×Mマトリクス、(2)三つの一次元LUT、(3)三次元LUT、(4)色空間変換、(5)色相シフト(Shift Hues) (カラーワーピング(color warping))、(6)ビジネスグラフィックス変換、(7)三線形(tri-linear)または角錐(pyramidal)の三次元LUTである。希望に応じて、このリストに対して、色変換演算を追加または削除することが可能である。

【0106】上記の色変換演算のそれぞれは、入力カラー画像データに異なる演算を行う。例えば、N×Mマトリクスは、カラー画像データをXYZ色空間からRGB色空間へ変換するために使用される3×3マトリクスであってもよいだろう。シーケンス構造体からアクセスされる他の演算に従い、同様の色変換を行うことができるだろう。

【0107】CMM9は、上記の七つの演算のリストを格納している。CMM9が、シーケンス構造体の「oper」値に基づき、カラー画像データに対してそれらの演算の一つを実行すべきであると判定すると、CMM9は、実行すべき演算に関するデータをその色変換演算を指定するプライベートタグから読み取り、そのデータを使用してカラー画像に対し色変換演算を実行する。次に、指定されたプライベートタグから、実行すべき演算に関するデータへのアクセスについて説明する。

【0108】すなわち、図10Bによれば、色変換演算それぞれの構造体をシーケンス構造体から読み取り、それに基づいて、どの種類の色変換演算を実行すべきかを判定する。例えば、CMM9は、メモリ内の領域を指示するポインタとして働く「oper」値として、特定の色変換演算を表す「1」を読み取った場合、「1」に相当する色変換演算、この場合はN×Mマトリクスにアクセスする。図9は、色変換演算と、シーケンス構造体66との関係をさらに示し、すなわち、色変換演算はシーケンス構造体66より階層が低くなっている。

【0109】図14は、UCCMS演算1、つまりN×Mマトリクスを実行するために使用されるデータを含むN×Mタグ80

の一例を示す。図14に示すように、 $N \times M$ マトリクスを実現するために使用されるプライベートタグのフォーマットは、ucmp62を実現するために使用されるプライベートタグのフォーマットに類似している。従って、簡潔にするため、ここではフォーマットの詳細な説明を省く。なお、色変換演算に使用されるプライベートタグそれぞれのフォーマットおよび構造は、 $N \times M$ タグ80と同じである。

【0110】図14に示す $N \times M$ タグ80は次を含む。つまり、前述したように、プライベートタグにより実行される色変換演算の種類を定義する「演算ID(Operation ID)」と、以下にさらに詳しく説明するが、この場合は使用されない「サブID(Sub ID)」と、後述するパラメータリストの長さを定義し、パラメータリストのアクセスに使用される「パラメータリストのバイト長(Length of Parameter List in bytes)」と、以下にさらに詳しく説明するが、この場合は使用されないタグフラグと、色変換演算で使用するべきパラメータを定義する「パラメータデータ(Parameter Data)」とである。図14に示すように、この場合、 N と M がともに3である $N \times M$ マトリクスにおいて使用されるパラメータは、マトリクスの N 次元、マトリクスの M 次元、および、各マトリクス要素である。

【0111】従って、CMM9は、図10Bに示す流れ図のステップS1006に基づき、 $N \times M$ の色変換演算を実行すべきであると判定した場合、 $N \times M$ タグ80に格納されているパラメータデータを読み取り、ステップS1007で、読取ったパラメータに従い「演算ID(Operation ID)」により定義される色変換演算を実行する。

【0112】シーケンス構造体1で定義される色変換演算の「oper」値が2である場合、処理はステップS1008へ進んだ後、ステップS1009へ進む。ステップS1008およびS1009は、一次元LUTを含む色変換演算を定義する。図15Aおよび図15Bは、「oper」値が2である色変換演算2のためのLUTを作成するために使用できると考えられるプライベートタグの二例を示している。

【0113】図15Aは色変換演算2のためのLUTを定義する一次元LUT(1DLUT)タグ81を示し、一次元LUT81からパブリックタグまたは別のプライベートタグを呼出す。さらに詳しくは、図15Aにおいて、「タグフラグ(Tag Flag)」値は「A2B2」である。「タグフラグ(Tag Flag)」は別のタグ、この場合は「AtoB2」パブリックタグの呼出しを定義する。すなわち、CMM9は、一次元LUT81を読み取り、「タグフラグ(TagFlag)」の値として「A2B2」を得ると、「AtoB2」パブリックタグをアクセスし、それに従って色変換演算を実行する。従って、この場合は、パブリックタグに格納されたデータに従い色変換演算を行うことになるので、「パラメータデータ(Parameter Data)」は不要である。そのため、「パラメータリストの長さ(Length of Parameter List)」もゼロ値を有する。

【0114】本発明によれば、ユーザが「サブID(Sub ID)」値を設定することにより、「タグフラグ(Tag Flag)」により指定されたパブリックタグ中の特定の色変換演算を選択できるので有利である。すなわち、「AtoB2」のようなパブリックタグはそれぞれ、二つ以上の色変換演算を含んでいてもよい。本発明は、パブリックタグから一つの色変換演算を選択する。例えば、パブリックタグ「AtoB2」は 3×3 マトリクス並びに三つのLUT、つまり、lut8Type/lut16Typeからの入力LUT、lut8Type/lut16Typeからの出力LUTおよび曲線タイプ(curve-type)からのLUTを含む。一次元LUTタグ81において「サブID(Sub ID)」値を0、1または2に設定することにより、CMM9は、パブリックタグ「AtoB2」で規定されたカラーマッピング演算のうち一つだけを実行する。さらに、CMM9は、一次元LUTタグ81の「演算ID(Operation ID)」に基づき、LUT演算を実行すべきであることを知るので、パブリックタグ「AtoB2」に格納された一つのLUTを選択するだけであり、例えば 3×3 マトリクスを選択することはない。

【0115】本発明の好ましい実施形態においては、「サブID(Sub ID)」の値「0」は、lut8Type/lut16Typeからの入力LUTを使用することをCMMに指示し、「サブID(Sub ID)」の値「1」は、lut8Type/lut16Typeからの出力LUTを使用することをCMM9に指示し、「サブID(Sub ID)」の値「2」は、曲線タイプ(curve-type)からのLUTを使用することをCMM9に指示するが、先に示したように、これらはすべてパブリックタグ「AtoB2」に格納されている。図4に示す各パブリックタグ、および、複数の色変換演算を含むと思われる他のパブリックタグについても、同様の指定を行うことができる。

【0116】図15Bは、「タグフラグ(Tag Flag)」がゼロ、すなわち、パブリックタグの情報に従ってではなく、プライベートタグの情報に従って色変換演算が実行されるべき、色変換演算2のための一次元LUTタグ82の一例を示している。すなわち、一次元LUTタグ82においては、前述したように、一次元LUTタグ82の16バイト目以降に記憶されている「パラメータデータ(Parameter Data)」のバイト長を指示する「パラメータリストの長さ(Length of Parameter List)」は77110である。一次元LUTタグ82で定義される「パラメータデータ(Parameter Data)」は、LUTを使用するガンマ補正のような色変換演算を実行するために必要なパラメータデータを含む。すなわち、図15Bに示すように、「パラメータデータ(Parameter Data)」は、LUTの次元、入出力データの入出力ビット数、および、LUTの値を含む。前述したように、パラメータの数、つまりLUTのサイズは、プライベートタグによって制限されない。これらの値は、本発明を実現するコンピュータ装置の能力によってのみ限定される。

【0117】図10Bに戻り、シーケンス構造体1で定義される色変換演算の「oper」値が3である場合、処理はス

ステップS1010へ進んだ後、ステップS1011へ進む。ステップS1010およびS1011は三次元LUTの色変換演算を定義する。図16は「oper」値が3である色変換演算3を実現するプライベートタグの一例を示す。図16の説明は、図15に関する先の説明と同様であるので、簡潔にするために図16の詳細な説明を省く。

【0118】図10Bに戻り、シーケンス構造体1で定義される色変換演算の「oper」値が4である場合、処理はステップS1012へ進んだ後、ステップS1013へ進む。ステップS1012およびS1013は色空間変換演算を定義する。

【0119】図17は「oper」値が4である色変換演算4を実現するプライベートタグ（色空間タグ84）の一例を示す。図17において、「演算ID(Operation ID)」は4、「パラメータリストの長さ(Length of Parameter List)」は0、「タグフラグ(Tag Flag)」は0Hである。この場合、これらの値は変化しない。この場合の「サブID(Sub ID)」は、色空間タグ84を介して実行されるべき色空間変換演算の種類を定義する。色空間タグ84を介して、1から9までの「サブID(Sub ID)」値により定義される次の色空間変換を実行できる。

- 0: RGB (赤、緑、青) →; HLS (色相、明度、彩度)
- 1: HLS →; RGB
- 2: CMY (シアン、マゼンタ、イエロー) →; HLS
- 3: HLS →; CMY
- 4: CIE XYZ →; CIE Lab
- 5: CIE Lab →; CIE XYZ
- 6: CIR xyY →; CIE XYZ
- 7: CIE XYZ →; CIE xyY
- 8: CIE XYZ →; CRGB (キヤノンRGB)
- 9: CRGB →; CIE XYZ

【0120】なお、上記の色空間変換のリストをすべて網羅しているわけではない、また、希望に応じて、別の色空間変換演算を追加してもよい。

【0121】シーケンス構造体1で定義される色変換演算の「oper」値が5である場合、図10Bの処理はステップS1014へ進んだ後、ステップS1015へ進む。ステップS1014およびS1015はカラーワーピング変換演算を定義する。簡単にいえば、カラーワーピング変換演算は、入力RGBの色相角を用いて、出力カラー画像データのCMYK誤差を修正するものである。

【0122】図18は「oper」値が5である色変換演算5を実現するワーピングタグ88の一例を示す。図18において、「演算ID(Operation ID)」は5、「サブID(Sub ID)」は0、「タグフラグ(Tag Flag)」は0Hである。これらの値は変化しない。この場合、「パラメータデータ(Parameter Data)」は、カラーワーピング変換関数を実行するために必要な値、すなわち、RGB色相角を含む。

【0123】シーケンス構造体1で定義される色変換演算の「oper」値が6である場合、図10Bの処理はステップS1016へ進んだ後、ステップS1017へ進む。ステップS101

6およびS1017は、例えば、現在係属中である米国特許出願番号第08/496,100号、名称「Color Management System Having Business Graphics Rendering Mode」に記載されているような色変換演算を実行するときに有用であるビジネスグラフィクスLUTを定義する。なお、この米国出願の内容は、参考として、本出願に取り入れられている。図19Aおよび図19Bに示すビジネスタグ89および90は、図15Aおよび図15Bに示す一次元LUTタグ81および82にそれぞれ対応している。

【0124】先の図15に示すプライベートタグに関する場合と同じで、図19に示すプライベートタグはucmP 62のシーケンス構造体1を介してCMM9によりアクセスされる。ところが、図11に示すプライベートタグに従って処理すべき入力データが同じ値を有する場合、CMM9は、図19に示すプライベートタグの何れかに従う色変換演算を実行せず、色変換演算は全く実行されない。これが、図19に示すプライベートタグに従って実行される色変換と、図15のそれとの唯一の相違点であるので、簡潔にするため、図19に示すプライベートタグにより定義される色変換演算の詳しい説明を省く。

【0125】シーケンス構造体1で定義される色変換演算の「oper」値が7である場合、図10Bの処理はステップS1018へ進んだ後、ステップS1019へ進む。ステップS1018およびS1019は、三次元線形補間を実行する三線形(tri-linear)または角錐(pyramidal)補間プロセスを定義する。

【0126】本発明の好ましい実施形態においては、三線形または角錐補間を実行するプライベートタグは、上述した他のプライベートタグに類似する構成を有する。図20は色変換演算7を実行するために使用されるプライベートタグ92の好ましい実施形態を示す。図12に示すように、プライベートタグ92において「演算ID(Operation ID)」は7、「サブID(Sub ID)」は0、「タグフラグ(Tag Flag)」はゼロである。これらの値は変化しない。さらに、「パラメータリストの長さ(Length of Parameter List)」はパラメータを構成するバイト数に相当する値を有し、三線形または角錐補間を実行するためのデータを定義する「パラメータデータ(Parameter Data)」も含まれている。

【0127】本発明に従って三線形または角錐補間を実行するために使用されるプライベートタグの別の実施形態を図21に示す。すなわち、図21に示すプライベートタグ39は、必要な色変換演算を実行するために使用するパラメータデータを含む代わりに、「テーブルID(Table ID)」を含む。「テーブルID(Table ID)」値は、三線形補間または角錐補間の何れを実行すべきかを判定するとき使用される別のプライベートタグを指す。この別のプライベートタグの一例を図22に示す。

【0128】さらに詳しくは、図22は、図21に示すプライベートタグ93から呼出され、三線形または角錐補間を

実行するために使用されるテーブルを定義するプライベートタグ94の一例を示している。

【0129】すなわち、図22に示すように、以下でucmT 94で示されるプライベートタグ94は、本発明における他のプライベートタグと同様に構成され、以下の要素を含む。つまり、タグ特有の識別子であり、CMM9がこのプライベートタグをアクセスする際に使用できる「uccms プライベート補間テーブルタグシグネチャ(uccmsPrivateInterpolation TableTag Signature)」と、図8に関して先に説明した「キャノンシグネチャ(Canon Signature)」と、ucmT 94のLUTの数を定義する「テーブル数(Number of Tables)」と、CMM9が三線形または角錐補間を実行する際のLUTである「テーブル(Table)」N (N=1, 2, ...) とである。

【0130】「テーブル(Table)」値は、図23に一例を示すテーブルヘッダ構造体を有する。図23に示すように、テーブルヘッダ構造体は次を含む。つまり、テーブル特有の識別子である「table (テーブルID)」と、指定されたテーブルの長さを定義する「length (長さ)」と、指定されたテーブルが始まるメモリの位置を定義する「offset (オフセット)」とである。この情報を使用して、メモリ内の異なるテーブルをアクセスすることができる。図9に示すように、これらのテーブルは色変換演算7からだけアクセスされる。

【0131】ucmT 94によりアクセスされるLUTを定義するプライベートタグの一例を図24に示す。図24に示すLUTプライベートタグ96は、本発明で使用する他のプライベートタグと同一のフォーマットを有する。

【0132】LUTプライベートタグ96は次の要素を含む。つまり、LUTにより実行される演算の種類、すなわち、その値が1ならば三線形補間、その値が0ならば角錐補間を定義する「テーブル種(tableKind)」と、LUTの親タグを定義する「親タグ(parentTag)」と、入力チャンネルの数を定義する「インチャンネル(inChannels)」と、LUTをアドレスするために使用される入力ビットの数を定義する「インビット(inBits)」と、出力チャンネルの数を定義する「アウトチャンネル(outChannels)」と、出力に使用されるビットの数、通常、この数と「インビット(inBits)」とはともに8である、を定義する「アウトビット(outBits)」と、LUTの格子点(grid points)の数を定義する「格子点三次元テーブル(gridPoints3DTable)」と、第四のチャンネルに関する格子点の数を定義し、角錐補間のときにのみ使用される「格子点四次元テーブル(gridPoints4DTable)」とである。上述したように、上記の情報は、三線形または角錐補間を実行するためにカラーマッチング法によって使用される。

【0133】「親タグ」が0x00000000であり、かつ、「テーブル種」が三線形であれば、ucmT 94によって完全補間カラーLUTが指される。「親タグ」が0x00000000であり、かつ、「テーブル種」が角錐であれば、カラー

LUTの中の第一のデータセットが主格子点データとして定義され、これに副交点データが続く。「親タグ」がプロファイル8の三次元LUTのシグネチャであり、かつ、「テーブル種」が三線形であれば、三次元LUTを格子点として使用し、「入力チャンネル」のような残りのフィールドは無視される。「親タグ」がプロファイル8の三次元LUTのシグネチャであり、かつ、「テーブル種」が角錐であれば、三次元LUTを主格子点として使用し、カラーLUTは副格子点のみを含む。

【0134】図10Bに戻り、ステップS1006からS1018の動作がいずれも実行されない場合、処理は終了する。なお、加えて、「プライベートデータテーブルタグ(PrivateDataTableTag)」と呼ぶプライベートデータタグは、ユーザによりユーザのプライベートデータを含むように変更され、階層記憶構造の中のあるポイントに格納されることができる。

【0135】次に、図25を参照して、本発明に従い、カラーモニタ2のRGB色空間からカラープリンタ16のCMY色空間へのカラー画像データの変換を簡単に説明する。なお、そのような色変換処理は、おそらく、現在利用できるパブリックタグを使用して実行可能であろうが、本発明を例示するために、以下の例ではユーザが作成したプライベートタグに関連して説明する。

【0136】カラーモニタ2とカラープリンタ16の双方について、定義済みのICCプロファイルが存在すると仮定すると、ステップS2501で、ユーザはプライベートタグを作成および変更するための上述のアプリケーションプログラム10を実行する。このプログラムに従い、ユーザは、RGB色空間からカラーモニタ2用のプロファイル結合空間へ、カラー画像データを変換するためのプライベートタグ情報と、そのプロファイル結合空間からカラープリンタ16のCMY色空間へ、カラー画像データを変換するためのプライベートタグ情報とを入力する。

【0137】ステップS2501でカラーモニタ2およびカラープリンタ16の双方についてプライベートタグを作成したならば、処理はステップS2502へ進む。ステップS2502で、CMM 9はカラーモニタ2のRGBカラー画像データを入力する。その後、ステップS2503で、CMM 9は、RGB色空間からプロファイル結合空間へカラー画像データを変換するためにICCプロファイルにアクセスする。

【0138】CMM 9は、カラーモニタ2用のカラープロファイルにアクセスしたならば、プロファイルにリストされているタグの何れをアクセスするかを決定するために、プロファイルを読取る。この場合、CMM 9は、CMMタイプ(CMMType)の下で「UCCM」を読取るであろうが、これは処理を、標準のICC手続に従わずに、ucmIプライベートタグに従い進めることを示す。

【0139】この例で使用されるICCプロファイルは、先に説明したのと同じ構造を有することに注意しなければならない。従って、ステップS2504で、CMM 9は、パブ

リックおよび/またはプライベートタグをRGBデータに作
用させるべきかを決定するために、ucmIを読取る。この
場合、ucmIのオーバーライド領域に非ゼロ値があると仮定
すると、CMM 9は、ucmPのシーケンス構造体領域の一つ
に同じ非ゼロ値がある場所をつきとめる。

【0140】従って、上記のステップにより、CMM 9
は、RGBカラー画像データに施すべき色変換演算のシー
ケンスを、すなわち、ucmIのオーバーライド領域の非ゼロ
値に対応するシーケンス構造体を介してアクセスされる
色変換演算を決定する。

【0141】CMM 9は、RGBカラー画像データに施される
べき色変換演算のシーケンスを決定すると、「oper」値
に基づいて、各色変換演算のデータおよび各色変換演算
のシーケンス構造体内の記憶位置データにアクセスす
る。このようにして、CMM 9は、 $N \times M$ マトリクスの要素
のような色変換演算データを格納するプライベートタグ
にアクセスする。

【0142】次に、ステップS2505で、CMM 9は、シーケ
ンス構造体により参照されたプライベートタグからのパ
ラメータを使用して、シーケンス構造体の「oper」値に
より示される色変換演算をRGBカラー画像データに施
す。この場合、RGB色空間スペースからプロファイル結
合空間へカラー画像データを変換するように、パラメー
タと色変換演算は設定されるであろう。

【0143】プロファイル結合空間への変換に続き、処
理はステップS2506へ進む。ステップS2506で、CMM 9は
カラープリンタ16のICCプロファイルにアクセスする。
なお、カラープリンタ16のICCプロファイルおよびカラ
ーモニタ2のICCプロファイルを、ディスク7のような一
つのメモリに格納させることも、別々のメモリに記憶さ
せることも可能であるが、一つのメモリに記憶させると
CMM9によるアクセスが容易になる。

【0144】CMM 9は、カラープリンタ16のカラープロ
ファイルにアクセスすると、カラーモニタ2のICCプロフ
ァイルに関して先に説明した場合と同じように、そのプロ
ファイル中にリストされたタグの何れにアクセスする
か決定するために、そのプロファイルを読取る。この場
合、CMMは、CMMタイプ(CMMType)の下で、標準のICC手続
に従わずに、ucmIプライベートタグに従って処理を進め
ることを示す「UCCM」を読取るだろう。

【0145】すなわち、前述したように、ステップS250
6からS2508において、CMM 9は、プロファイル結合空間
からカラープリンタ16のCMY色空間へカラー画像データ
を変換するために、カラープリンタ16用のICCプロフ
ァイルのucmIおよびucmPプライベートタグをアクセスする
だろう。この処理の概要は、カラーモニタ2用のICCプロ
ファイルに関して先に説明した処理と同じであるので、
簡潔にするために、ここでは処理の詳細な説明を省く。

【0146】ステップS2508に続き、処理はステップS25
09へ進み、色変換したCMYカラー画像データをプリンタ1

6へ出力する。

【0147】なお、ICCプロファイルフォーマットに関
連して本発明を説明したが、色変換演算を格納できアク
セスできるどのような種類のクロスプラットフォームデ
バイスのフォーマットと組み合わせても、本発明を使用
することができる。

【0148】同様に、カラーモニタおよびカラープリン
タに関連して本発明を説明したが、カラーキャナまた
はカラーファクシミリ装置などのような他のカラー画像
処理デバイスと組み合わせても、本発明を利用すること
ができる。

【0149】特定の一実施形態に関連して本発明を説明
した。本発明は、上述した実施形態には限定されず、当
業者により特許請求の範囲の趣旨から逸脱することなく
様々な変更および変形を実施し得る。

【0150】

【他の実施形態】なお、本発明は、複数の機器（例えば
ホストコンピュータ、インタフェイス機器、リーダ、プ
リントなど）から構成されるシステムに適用しても、一
つの機器からなる装置（例えば、複写機、ファクシミリ
装置など）に適用してもよい。

【0151】また、本発明の目的は、前述した実施形態
の機能を実現するソフトウェアのプログラムコードを記
録した記憶媒体を、システムあるいは装置に供給し、そ
のシステムあるいは装置のコンピュータ（またはCPUやM
PU）が記憶媒体に格納されたプログラムコードを読み
実行することによっても、達成されることは言うまでも
ない。この場合、記憶媒体から読出されたプログラムコ
ード自体が前述した実施形態の機能を実現することにな
り、そのプログラムコードを記憶した記憶媒体は本発明
を構成することになる。プログラムコードを供給するた
めの記憶媒体としては、例えば、フロッピーディスク、ハ
ードディスク、光ディスク、光磁気ディスク、CD-ROM、
CD-R、磁気テープ、不揮発性のメモリカード、ROMなど
を用いることができる。

【0152】また、コンピュータが読出したプログラム
コードを実行することにより、前述した実施形態の機能
が実現されるだけでなく、そのプログラムコードの指示
に基づき、コンピュータ上で稼働しているOS（オペレー
ティングシステム）などが実際の処理の一部または全部
を行い、その処理によって前述した実施形態の機能が実
現される場合も含まれることは言うまでもない。

【0153】さらに、記憶媒体から読出されたプログラ
ムコードが、コンピュータに挿入された機能拡張カード
やコンピュータに接続された機能拡張ユニットに備わる
メモリに書込まれた後、そのプログラムコードの指示に
基づき、その機能拡張カードや機能拡張ユニットに備わ
るCPUなどが実際の処理の一部または全部を行い、その
処理によって前述した実施形態の機能が実現される場合
も含まれることは言うまでもない。

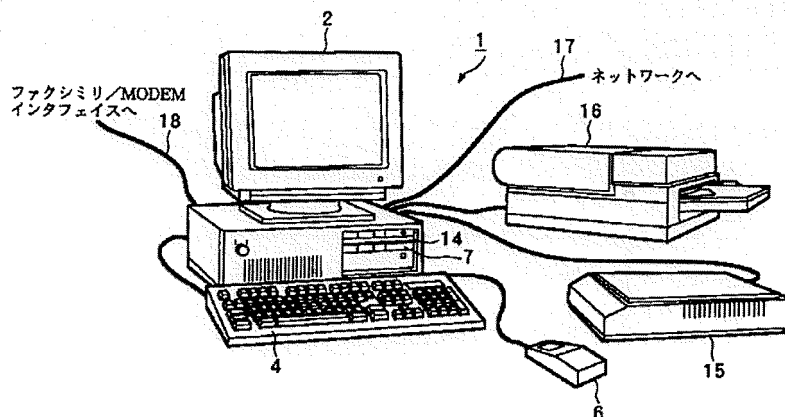
【図面の簡単な説明】

【図 1】本発明の動作において使用されるコンピュータハードウェアの見通し図、
 【図 2】本発明のブロック図、
 【図 3】デバイス従属色空間とプロファイル結合空間との関係を示す図、
 【図 4】ICCプロファイルを示す図、
 【図 5】プライベートタグの作成および変更を示す流れ図、
 【図 6】プライベートタグ情報を描写する画面を示す図、
 【図 7】プライベートタグ情報を描写する画面を示す図、
 【図 8】ucmIプライベートタグを示す図、
 【図 9】本発明により作成されるプライベートタグの階層関係を示す図、
 【図 10 A】本発明に従う色変換処理のセットアップを示す流れ図、
 【図 10 B】本発明に従う色変換処理のセットアップを示す流れ図、
 【図 11】ucmPプライベートタグを示す図、
 【図 12】シーケンス構造体を示す図、
 【図 13】色変換演算の構造体を示す図、
 【図 14】N×Mマトリクスのプライベートタグを示す

図、

【図 15 A】三つの一次元ルックアップテーブルのプライベートタグ例を示す図、
 【図 15 B】三つの一次元ルックアップテーブルのプライベートタグ例を示す図、
 【図 16】三次元ルックアップテーブルのプライベートタグを示す図、
 【図 17】色空間のプライベートタグを示す図、
 【図 18】カラーワーピングのプライベートタグを示す図、
 【図 19 A】ビジネスグラフィックスルックアップテーブルのプライベートタグ例を示す図、
 【図 19 B】ビジネスグラフィックスルックアップテーブルのプライベートタグ例を示す図、
 【図 20】三線形補間または角錐補間のプライベートタグの第一の実施形態を示す図、
 【図 21】三線形補間または角錐補間のプライベートタグの第二の実施形態を示す図、
 【図 22】ucmTプライベートタグを示す図、
 【図 23】テーブル構造体を示す図、
 【図 24】テーブルのプライベートタグを示す図、
 【図 25】本発明を使用して、ある色空間から別の色空間へ入力画像データを変換する処理ステップを示す流れ図である。

【図 1】

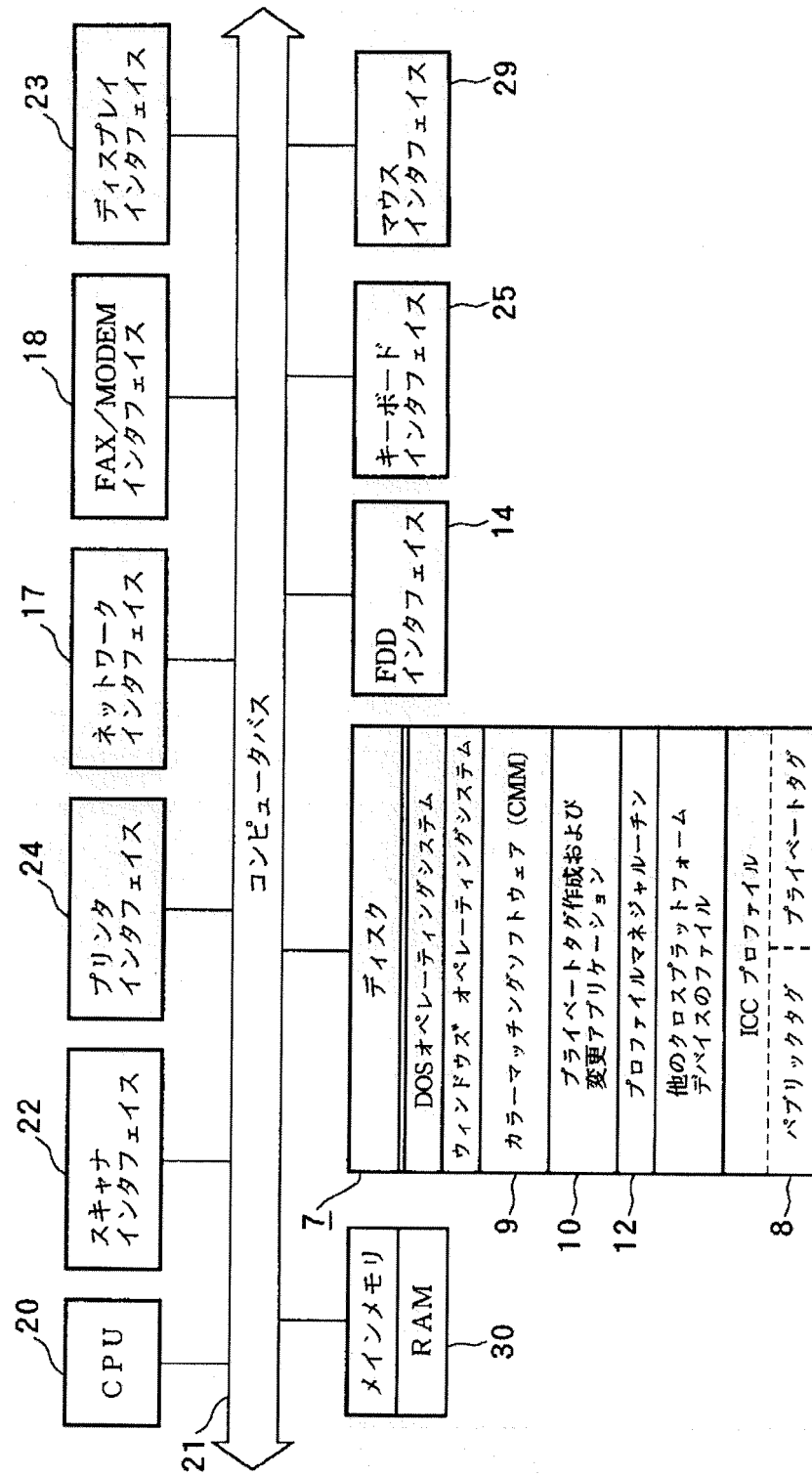


【図 12】

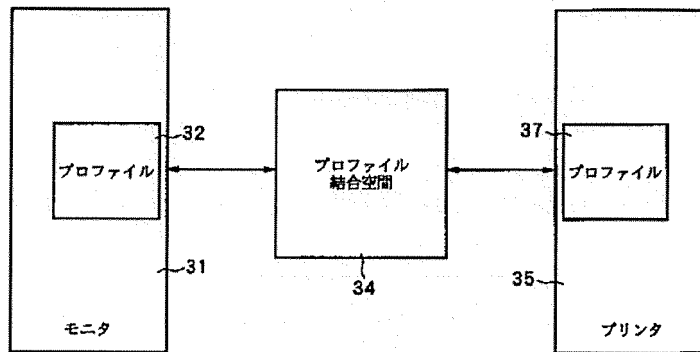
```

/*UCCMSシーケンス構造体*/
typedef struct {
    uint32    33      CanonID;    /*シーケンス構造体用のUCCMS ID*/
    uint32    00      reserved; /*常に0*/
    uint32    04      numOps;    /*シーケンスの演算数*/
    uint32    LAB     seqPCS;    /*このシーケンスのプロファイル結合空間*/
    uint32    40      length;    /*演算タグのバイト長*/
    uint32    5163    offset;    /*ucmPの演算タグ開始オフセット*/
}
    
```

【図2】



【図3】



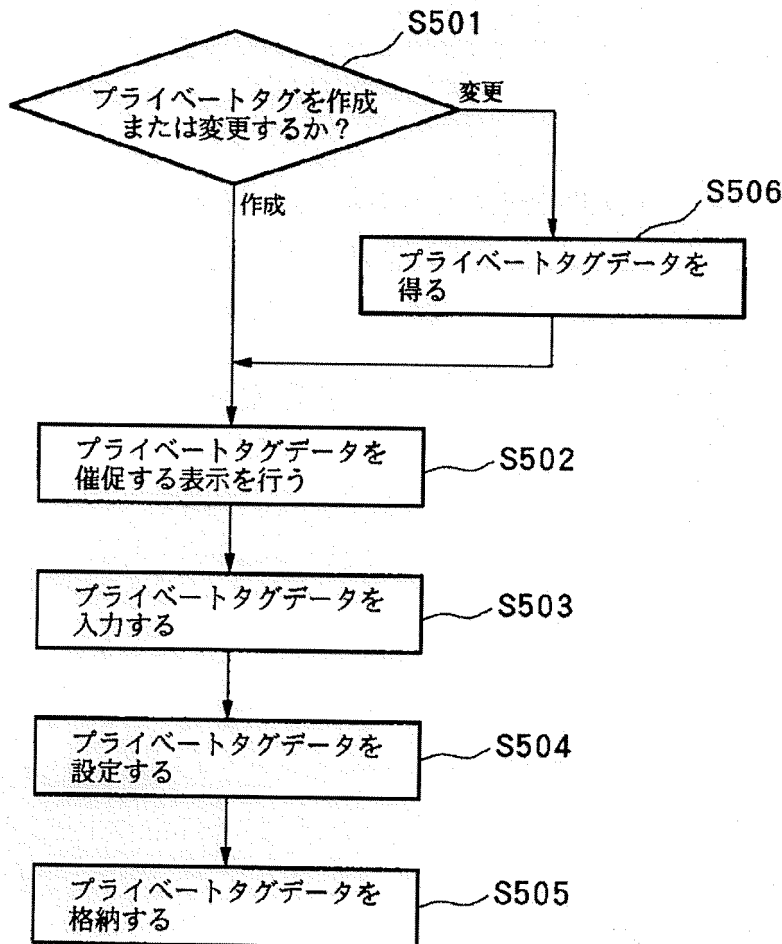
【図4】

template.prf				ヘッダ			
サイズ: 258954 バイト CMMタイプ: "UCCM" バージョン: 0x02000000 プロファイルクラス: "prtr" データ色空間: "CMY" インタチェンジ空間: "Lab" 作成日: 23.6.95, 16:51:8 CS2 シグネチャ: "acsp"				一次プラットフォーム: "MSFT" フラグ: 0x00000000 デバイス製造者: "CANO" デバイスモデル: 1110716737 デバイス属性: 0x00000204, 0x01000000 レンダリング特性: 0 白色XYZ: X = 0.9842, Y = 1.0000, Z = 0.8248			
タグテーブル (19 elements, double-click to inspect)							
Ind	Signal	エレメントオフセット	サイズ	Ind	Signal	エレメントオフセット	サイズ
0	'meas'	360 = 0x00000168	20	16	'A2B2'	147283 = 0x00023F63	108395
1	'cpri'	380 = 0x0000017C	82	17	'ucal'	256678 = 0x0003EAA6	240
2	'dmnd'	462 = 0x000001CE	115	18	'ucm1'	256916 = 0x0003EB86	2436
3	'dmdd'	577 = 0x00000241	123				
4	'wtp1'	700 = 0x000002BC	20				
5	'tech'	720 = 0x000002D0	12				
6	'desc'	732 = 0x000002DC	147				
7	'pre0'	879 = 0x0000036F	108395				
8	'pre1'	879 = 0x0000036F	108395				
9	'pre2'	879 = 0x0000036F	108395				
10	'gam1'	110274 = 0x0001AEC2	37009				
11	'B2A0'	147283 = 0x00023F63	108395				
12	'B2A1'	147283 = 0x00023F63	108395				
13	'B2A2'	147283 = 0x00023F63	108395				
14	'A2B0'	147283 = 0x00023F63	108395				
15	'A2B1'	147283 = 0x00023F63	108395				

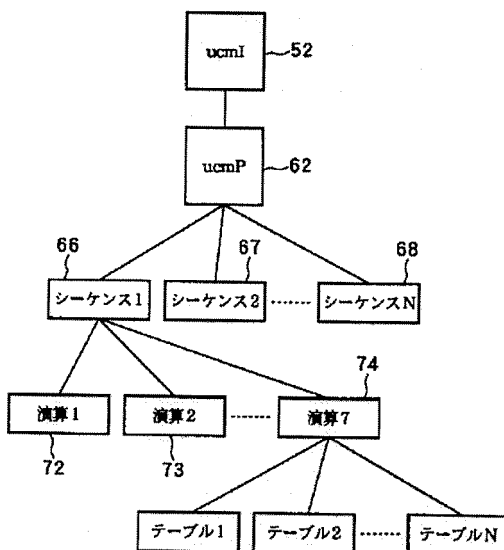
【図11】

バイト	記述	データ	データ型
0-3	uccmsプライベートシーケンスタグテーブルシグネチャ	'uomP'	icSignature
4-7	キヤノンシグネチャ	'CSIG'	uint32
8-11	シーケンス数	4*	uint32
12-15	予約	0x00	uint32
16-39	シーケンス構造体1		uccmsシーケンス構造体
40-63	シーケンス構造体2		uccmsシーケンス構造体
64-87	シーケンス構造体3		uccmsシーケンス構造体
88-111	シーケンス構造体4		uccmsシーケンス構造体

【図5】



【図9】



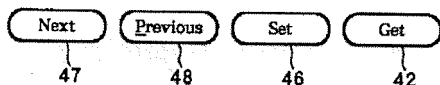
【図14】

バイト	記述	データ	データ型
0-3	演算ID	1	uint32
4-7	サブID	0x00	uint32
8-11	パラメータリストのバイト長	44*	uint32
12-15	タグフラグ (常にパラメータデータを使う)	0x00000000	uint32
16	パラメータデータ-n次元	0x03*	uint32
	パラメータデータ-m次元	0x03*	uint32
	パラメータデータ-マトリクス要素a1	1*	s15Fixed16
	パラメータデータ-マトリクス要素a2	0*	s15Fixed16
	パラメータデータ-マトリクス要素a3	0*	s15Fixed16
	パラメータデータ-マトリクス要素b1	0*	s15Fixed16
	パラメータデータ-マトリクス要素b2	1*	s15Fixed16
	パラメータデータ-マトリクス要素b3	0*	s15Fixed16
	パラメータデータ-マトリクス要素c1	0*	s15Fixed16
	パラメータデータ-マトリクス要素c2	0*	s15Fixed16
	パラメータデータ-マトリクス要素c3	1*	s15Fixed16

80

【図6】

プライベート情報タグ	
ucms プライベート情報タグシグネチャ	'ucml'
キャノンシグネチャ	'CSIG'
パラメータのバイトサイズ	244
エンジンバージョン	01000001
プロファイルフォーマット文書バージョン	01000001
プロファイル バージョン	01000001
プロファイルビルド番号	01000001
補間フラグ	01000001
AtoB0Tag オーバライド	0
AtoB1Tag オーバライド	0
AtoB2Tag オーバライド	0
BtoA0Tag オーバライド	0
BtoA1Tag オーバライド	0
BtoA2Tag オーバライド	33



43

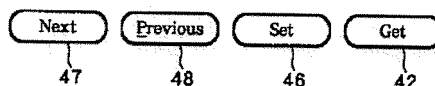
【図8】

バイト	記 述	データ	データ型
0-3	ucms プライベート情報タグシグネチャ	'ucml'	icSignature
4-7	キャノンシグネチャ	'CSIG'	uint32
8-11	パラメータのバイトサイズ	244*	uint32
12-15	エンジンバージョン	0x01000001*	uint32
16-19	プロファイルフォーマット文書バージョン	0x01010000*	uint32
20-23	プロファイルバージョン	0x01000000*	uint32
24-27	プロファイルビルド番号	0x00000001*	uint32
28-31	補間フラグ	0x00000001*	uint32
32-35	AtoB0Tag オーバライド	0x0*	uint32
36-39	AtoB1Tag オーバライド	0x0*	uint32
40-43	AtoB2Tag オーバライド	0x0*	uint32
44-47	BtoA0Tag オーバライド	0x0*	uint32
48-51	BtoA1Tag オーバライド	0x0*	uint32
52-55	BtoA2Tag オーバライド	0x33*	uint32
56-59	preview0Tag オーバライド	0x0*	uint32
60-63	preview1Tag オーバライド	0x0*	uint32
64-67	preview2Tag オーバライド	0x0*	uint32
68-71	gamutTag オーバライド	0x0*	uint32
72-75	AtoB0Tag 最適化フラグ	0x0*	uint32
76-79	AtoB1Tag 最適化フラグ	0x0*	uint32
80-83	AtoB2Tag 最適化フラグ	0x0*	uint32
84-87	BtoA0Tag 最適化フラグ	0x0*	uint32
88-91	BtoA1Tag 最適化フラグ	0x0*	uint32
92-95	BtoA2Tag 最適化フラグ	0x0*	uint32
96-99	preview0Tag 最適化フラグ	0x0*	uint32
100-103	preview1Tag 最適化フラグ	0x0*	uint32
104-107	preview2Tag 最適化フラグ	0x0*	uint32
108-111	gamutTag 最適化フラグ	0x0*	uint32
112-115	作成部門	CIS*	char[64]
116-119	サポート部門	CINC	char[64]
120-123	Von Kries フラグ	0x0*	uint32
...	Possible fields in the future		

52

【図7】

プライベート情報タグ	
preview0Tag オーバライド	0
preview1Tag オーバライド	0
preview2Tag オーバライド	0
gamutTag オーバライド	0
AtoB0Tag 最適化フラグ	0
AtoB1Tag 最適化フラグ	0
AtoB2Tag 最適化フラグ	0
BtoA0Tag 最適化フラグ	0
BtoA1Tag 最適化フラグ	0
BtoA2Tag 最適化フラグ	0
preview0Tag 最適化フラグ	0
preview1Tag 最適化フラグ	0
preview2Tag 最適化フラグ	0
gamutTag 最適化フラグ	0
作成部門	CIS
サポート部門	CINC
Von Kries フラグ	0



44

【図15A】

バイト	記 述	データ	データ型
0-3	演算ID	2	uint32
4-7	サブID	0*	uint32
8-11	パラメータリストのバイト長	0	uint32
12-15	タグフラグ	'A2B2'*	uint32

81

【図15B】

バイト	記 述	データ	データ型
0-3	演算ID	2	uint32
4-7	サブID	0	uint32
8-11	パラメータリストのバイト長	771*	uint32
12-15	タグフラグ	0	uint32
16	パラメータデータ-次元	0x03*	uint8
17	パラメータデータ-入力ビット数	0x08*	uint8
18	パラメータデータ-出力ビット数	0x08*	uint8
19	パラメータデータ-次元テーブル	0xFF*	uint8
	パラメータデータ...	--	--

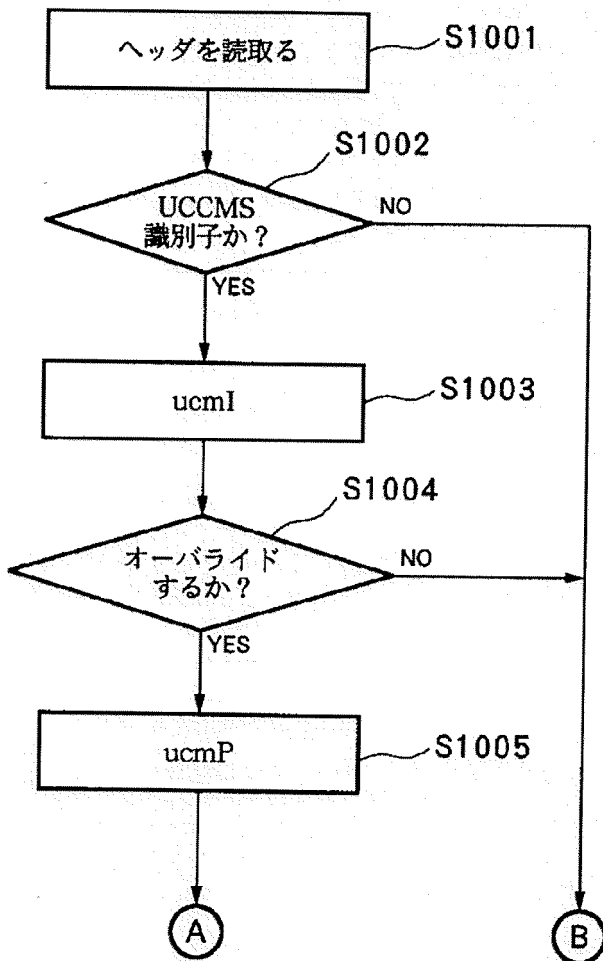
82

【図16】

バイト	記 述	データ	データ型
0-3	演算ID	3	uint32
4-7	サブID	0	uint32
8-11	パラメータリストのバイト長	1	uint32
12-15	タグフラグ	0	uint32
16-19	テーブルID	0x11*	uint32

83

【図10A】



【図13】

```

/*UCCMS演算構造体*/
typedef struct {
  UInt32      oper;      /*UCCMS演算番号*/
  UInt32      subid;     /*UCCMSサブID番号*/
  UInt32      length;    /*パラメータリストのバイト長*/
  UInt32      tagFlag;   /*フラグ、0x00ならばparmのデータを使い、*/
  UInt32      parm[icAny] /*そうでなければタグパラメータリストを使う*/
}
  
```

【図18】

バイト	記述	データ	データ型
0-3	演算ID	5	uint32
4-7	サブID	0	uint32
8-11	パラメータリストのバイト長	6	uint32
12-15	タグフラグ	0x00	uint32
16-	プリンタ赤 (m+y) 色相角	20*	uint32
	プリンタ緑 (c+y) 色相角	176*	uint32
	プリンタ青 (m+c) 色相角	300*	uint32

88

【図17】

バイト	記述	データ	データ型
0-3	演算ID	4	uint32
4-7	サブID	2*	uint32
8-11	パラメータリストのバイト長	0*	uint32
12-15	タグフラグ	0x00	uint32

84

【図19B】

バイト	記述	データ	データ型
0-3	演算ID	6	uint32
4-7	サブID	0	uint32
8-11	パラメータリストのバイト長	771*	uint32
12-15	タグフラグ	0	uint32
16	パラメータデータ- 次元	0x06*	uint8
17	パラメータデータ- 入力ビット数	0x06*	uint8
18	パラメータデータ- 出力ビット数	0x06*	uint8
19-	パラメータデータ- 次元テーブル	0xFF*	uint8
	パラメータデータ...

90

【図20】

バイト	記述	データ	データ型
0-3	演算ID	7	uint32
4-7	サブID	0	uint32
8-11	パラメータリストのバイト長	1	uint32
12-15	タグフラグ	0	uint32
16	パラメータデータ-	0x06*	uint8
17	パラメータデータ-	0x06*	uint8
18	パラメータデータ-	0x06*	uint8
19-	パラメータデータ-	0x06*	uint8
	パラメータデータ...

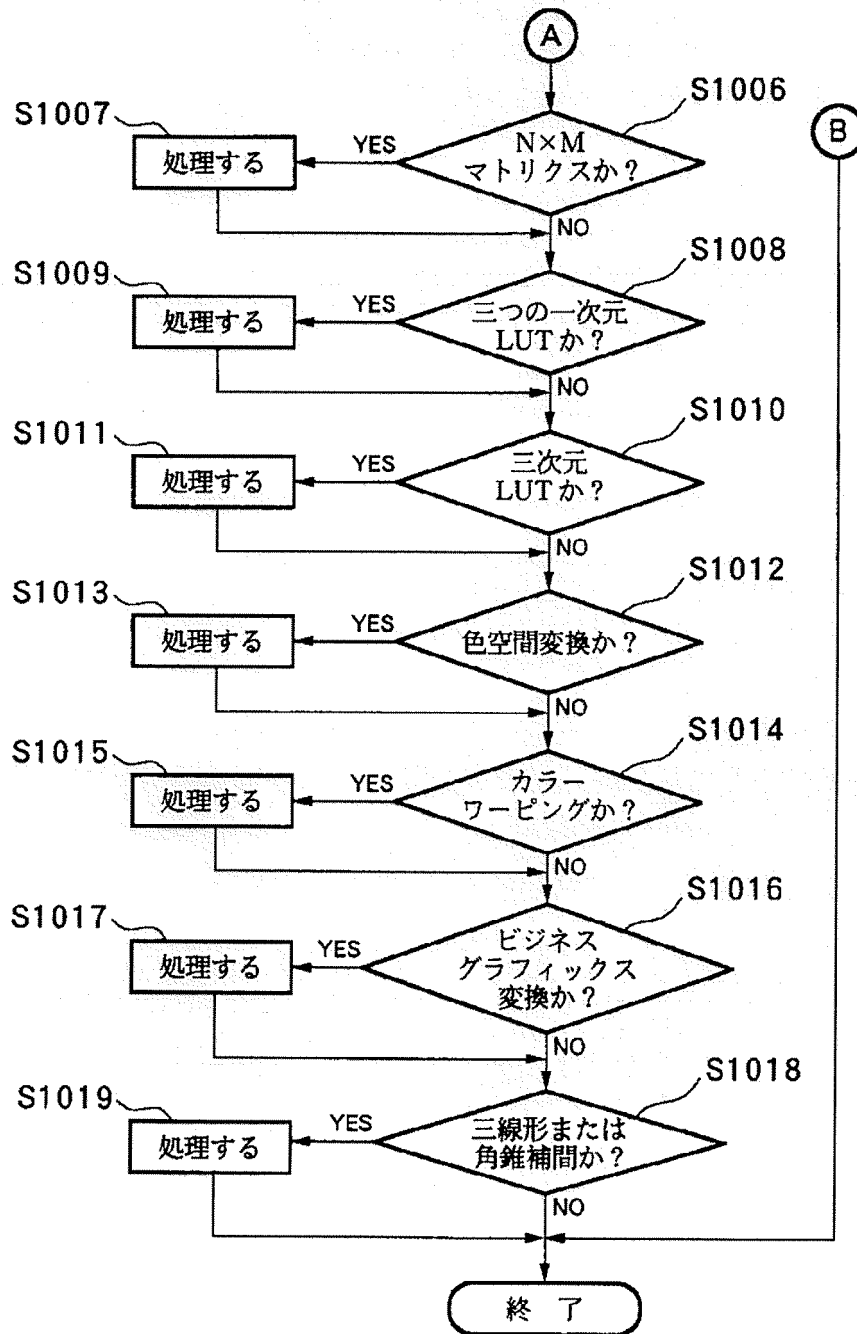
92

【図19A】

バイト	記述	データ	データ型
0-3	演算ID	6	uint32
4-7	サブID	0*	uint32
8-11	パラメータリストのバイト長	0	uint32
12-15	タグフラグ	A2B2*	uint32

89

【図10B】



【図21】

バイト	記述	データ	データ型
0-3	演算ID	7	uint32
4-7	サブID	0	uint32
8-11	パラメータリストのバイト長	1	uint32
12-15	タグフラグ	0	uint32
16-19	テーブルID	0x04*	uint32

【図 2 2】

バイト	記 述	データ	データ型
0-3	uccms プライベート補間テーブルタグシグネチャ	'ucmT'	icSignature
4-7	キャノンシグネチャ	'CSIG'	uint32
8-11	テーブル数	3*	uint32
12-15	予約	0	uint32
16-	テーブル1		uccmsIntrpLUTType
	テーブル2		uccmsIntrpLUTType
	テーブル3		uccmsIntrpLUTType

94 ↗

【図 2 3】

/*UCCMS補間テーブルヘッダ構造体*/

typedef struct {

uint32 tableID; /*テーブルのID番号*/

uint32 length; /*テーブルのバイト長*/

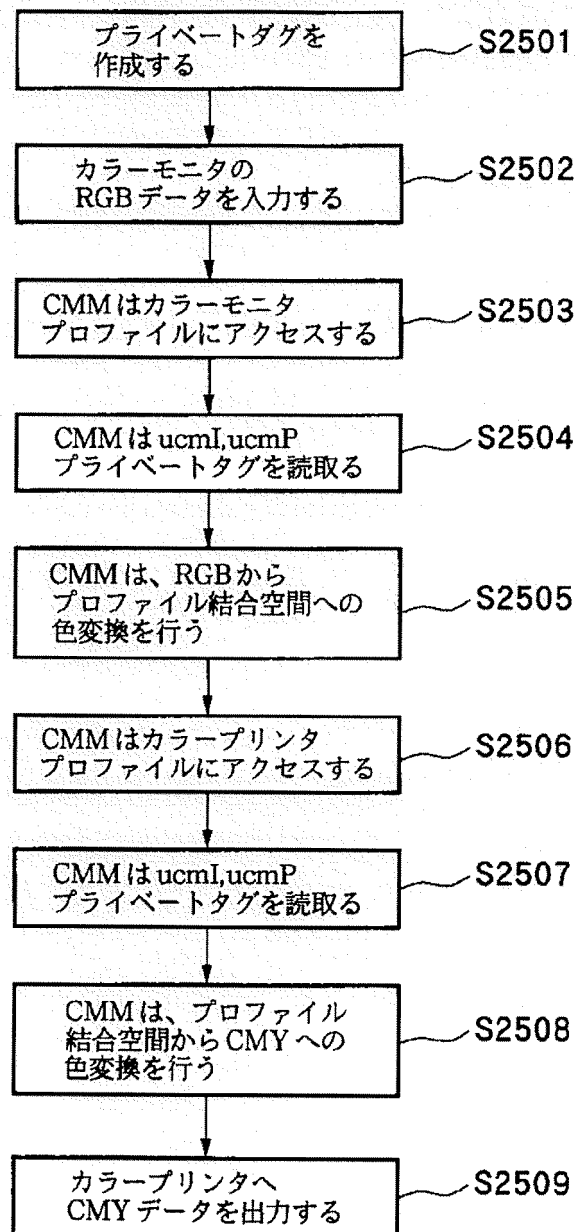
uint32 offset; /*ucmTのテーブル開始オフセット*/

【図 2 4】

バイト	記 述	データ	データ型
0-3	テーブル種	0*	uint32
4-7	親タグ	0x00*	icSignature
8	インチャネル	0x03*	uint8
9	インビット	0x08*	uint8
10	アウトチャネル	0x04*	uint8
11	アウトビット	0x06*	uint8
12	格子点三次元テーブル	0x21*	uint8
13	格子点四次元テーブル	0x21*	uint8
14-15	予約	0x00	uint16
16-	CLUT (カラー-LUT)		

96 ↗

【図25】



フロントページの続き

(51) Int. Cl.⁶
H 0 4 N 1/46

識別記号 庁内整理番号

F I
H 0 4 N 1/46

技術表示箇所

Z

(72) 発明者 ジョナサン ワイ、 ヒュイ
アメリカ合衆国 カリフォルニア州
94536, フレモント, カリストガ サ
ークル 559

【外国語明細書】

1. Title of Invention

COLOR IMAGE PROCESSING APPARATUS AND METHOD

2. Claims

(1) An apparatus for performing a color transformation operation on input image data, said apparatus using a first tag and a second tag to perform the color transformation operation in place of a predetermined series of color transformation operations stored in a public tag in a profile format, said apparatus comprising:

inputting means for inputting color image data;

a memory for storing the first tag and the second tag, the first tag for storing override information for overriding the predetermined series of color transformations, and the second tag for storing color transformation operation data accessible via a hierarchical storage structure;

a determining means for determining, based on the override information in the first tag, whether to access the second tag;

a reading means for reading the color transformation operation data in the second tag in a case that said determining means determines to access the second tag, the reading means following pointers within the hierarchical storage structure to access the color transformation operation data in the second tag; and

a processing means for performing a color transformation operation on the input image data in accordance with the color transformation operation data read by said reading means.

(2) An apparatus according to Claim 1, wherein the override

information stored in the first tag is used to access a color transformation sequence in the second tag, and wherein said reading means accesses the color transformation sequence based on the override information.

(3) An apparatus according to Claim 2, where the color transformation sequence stores a pointer to the color transformation operation, and wherein said reading means accesses the color transformation operation based on the pointer stored in the color transformation sequence.

(4) An apparatus according to Claim 1, wherein said storing means stores a list of color transformation operations and wherein said processing means accesses a color transformation operation in the list of color transformation operations based on the color transformation operation data read by said reading means.

(5) An apparatus according to Claim 1, wherein in a case that said determining means determines not to access the second tag, the input image data is processed in accordance with the predetermined series of color transformation operations stored in the public tag.

(6) A method for performing a color transformation operation on input image data, said method using a first tag and a second tag to perform the color transformation operation in place of a predetermined series of color transformation operations stored in a public tag in a profile format, said method comprising:

an inputting step for inputting color image data;

a storing step for storing the first tag and the second tag, the first tag for storing override information for

overriding the predetermined series of color transformations, and the second tag for storing color transformation operation data accessible via a hierarchical storage structure;

a determining step for determining, based on the override information in the first tag, whether to access the second tag;

a reading step for reading the color transformation operation data in the second tag in a case that said determining step determines to access the second tag, said reading step following pointers within the hierarchical storage structure to access the color transformation operation data in the second tag; and

a processing step for performing a color transformation operation on the input image data in accordance with the color transformation operation data read in said reading step.

(7) A method according to Claim 6, wherein the override information stored in the first tag is used to access a color transformation sequence in the second tag, and wherein said reading step accesses the color transformation sequence based on the override information.

(8) A method according to Claim 7, where the color transformation sequence stores a pointer to the color transformation operation, and wherein said reading step accesses the color transformation operation based on the pointer stored in the color transformation sequence.

(9) A method according to Claim 6, wherein said storing step stores a list of color transformation operations and wherein said processing step accesses a color transformation operation in the list of color transformation operations based on the

color transformation operation data read by said reading means.

(10) A method according to Claim 6, wherein in a case that said determining step determines not to access the second tag, the input image data is processed in accordance with the predetermined series of color transformation operations stored in the public tag.

(11) Computer-executable process steps stored on a computer-usable storage medium, said process steps for performing a color transformation operation on input image data, said process steps using a first tag and a second tag to perform the color transformation operation in place of a predetermined series of color transformation operations stored in a public tag in a profile format, said process steps comprising:

- an inputting step for inputting color image data;

- a storing step for storing the first tag and the second tag, the first tag for storing override information for overriding the predetermined series of color transformations, and the second tag for storing color transformation operation data accessible via a hierarchical storage structure;

- a determining step for determining, based on the override information in the first tag, whether to access the second tag;

- a reading step for reading the color transformation operation data in the second tag in a case that said determining step determines to access the second tag, said reading step following pointers within the hierarchical storage structure to access the color transformation operation data in the second tag; and

- a processing step for performing a color transformation

operation on the input image data in accordance with the color transformation operation data read in said reading step.

(12) Computer-executable process steps according to Claim 11, wherein the override information stored in the first tag is used to access a color transformation sequence in the second tag, and wherein said reading step accesses the color transformation sequence based on the override information.

(13) Computer-executable process steps according to Claim 12, where the color transformation sequence stores a pointer to the color transformation operation, and wherein said reading step accesses the color transformation operation based on the pointer stored in the color transformation sequence.

(14) Computer-executable process steps according to Claim 11, wherein said storing step stores a list of color transformation operations and wherein said processing step accesses a color transformation operation in the list of color transformation operations based on the color transformation operation data read by said reading means.

(15) Computer-executable process steps according to Claim 11, wherein in a case that said determining step determines not to access the second tag, the input image data is processed in accordance with the predetermined series of color transformation operations stored in the public tag.

(16) In a profile format in which numeric information is stored for use by a standardized transformation from one color space to another color space, and in which tags are hierarchically stored, an apparatus for using the tags to modify the standardized transformation, said apparatus comprising:

an inputting means for inputting color image data;

a memory (1) for storing first pointer data at predetermined locations in a first tag which is at a highest level in a hierarchy of tags, the first pointer data pointing to a color transformation sequence in a second tag, the second tag being at a second highest level in the hierarchy of tags, (2) for storing the color transformation sequence in the second tag, the color transformation sequence including a second pointer for pointing to a color transformation operation stored in a third tag, the third tag being at a third highest level in the hierarchy of tags, and (3) for storing data for performing the color transformation operation in the third tag;

a determining means for determining whether to access the first tag based on information stored in the profile format;

a first reading means (1) for reading, in the case that the first tag is to be accessed, the first pointer information in the first tag and determining whether to access the color transformation sequence, (2) for reading, in the case that the color transformation sequence is to be accessed, the second pointer in the color transformation sequence to determine whether to access the color transformation operation, and (3) for reading, in the case that the color transformation operation is to be accessed, the data for performing the color transformation operation from the third tag; and

a processing means for performing the color transformation operation in accordance with the data read in the data from the third tag to transform the input image data from one color space to another color space in place of the standardized

transformation.

(17) An apparatus according to Claim 16, wherein, in a case that the color transformation operation comprises a predetermined color transformation operation, the data for performing the color transformation operation comprises third pointer data for pointing to a fourth tag, the fourth tag being at a fourth level in the hierarchy of tags.

(18) An apparatus according to Claim 17, wherein said storing means stores table data in the fourth tag; and

wherein said processing means further performs the color transformation operation defined in the third tag in accordance with the table data stored in the fourth tag to transform the input image data from one color space to another color space in place of the standardized transformation.

(19) An apparatus according to Claim 16, wherein said determining means determines whether to access the first tag by reading CMM data from a header stored in the profile format and by searching the first tag for data corresponding to the CMM data.

(20) An apparatus according to Claim 16, wherein said processing means comprises color matching software, and wherein said processing means executes the color matching software in accordance with a user's input.

(21) An apparatus according to Claim 16, wherein said processing means determines which color transformation operation to perform by matching the data read by said reading means from the third tag to a pre-stored list of color transformation operations.

(22) In a profile format in which numeric information is

stored for use by a standardized transformation from one color space to another color space, and in which tags are hierarchically stored, a method for using the tags to modify the standardized transformation, said method comprising:

- an inputting step for inputting color image data;

- a first storing step for storing first pointer data at predetermined locations in a first tag which is at a highest level in a hierarchy of tags, the first pointer data pointing to a color transformation sequence in a second tag, the second tag being at a second highest level in the hierarchy of tags;

- a second storing step for storing the color transformation sequence in the second tag, the color transformation sequence including a second pointer for pointing to a color transformation operation stored in a third tag, the third tag being at a third highest level in the hierarchy of tags;

- a third storing step for storing data for performing the color transformation operation in the third tag;

- a determining step for determining whether to access the first tag based on information stored in the profile format;

- a first reading step for reading, in the case that the first tag is to be accessed, the first pointer information in the first tag and determining whether to access the color transformation sequence;

- a second reading step for reading, in the case that the color transformation sequence is to be accessed, the second pointer in the color transformation sequence to determine whether to access the color transformation operation;

- a third reading step for reading, in the case that the color transformation operation is to be accessed, the data for

performing the color transformation operation from the third tag; and

a performing step for performing the color transformation operation in accordance with the data read in said third reading step to transform the input image data from one color space to another color space in place of the standardized transformation.

(23) A method according to Claim 22, wherein, in a case that the color transformation operation comprises a predetermined color transformation operation, the data for performing the color transformation operation comprises third pointer data for pointing to a fourth tag, the fourth tag being at a fourth level in the hierarchy of tags.

(24) A method according to Claim 23, said method further comprising a fourth storing step for storing table data in the fourth tag; and

a second performing step for performing the color transformation operation defined in the third tag in accordance with the table data stored in the fourth tag to transform the input image data from one color space to another color space in place of the standardized transformation.

(25) A method according to Claim 22, wherein said determining step determines whether to access the first tag by reading CMM data from a header stored in the profile format and by searching the first tag for data corresponding to the CMM data.

(26) A method according to Claim 22, wherein said performing step determines which color transformation operation to perform by matching the data read in said third reading step to a pre-stored list of color transformation operations.

(27) In a profile format in which numeric information is stored for use by a standardized transformation from one color space to another color space, and in which tags are hierarchically stored, computer-executable process steps stored on a computer-readable medium for using the tags to modify the standardized transformation, said process steps comprising:

- an inputting step for inputting color image data;

- a first storing step for storing first pointer data at predetermined locations in a first tag which is at a highest level in a hierarchy of tags, the first pointer data pointing to a color transformation sequence in a second tag, the second tag being at a second highest level in the hierarchy of tags;

- a second storing step for storing the color transformation sequence in the second tag, the color transformation sequence including a second pointer for pointing to a color transformation operation stored in a third tag, the third tag being at a third highest level in the hierarchy of tags;

- a third storing step for storing data for performing the color transformation operation in the third tag; and

- a determining step for determining whether to access the first tag based on information stored in the profile format;

- a first reading step for reading, in the case that the first tag is to be accessed, the first pointer information in the first tag and determining whether to access the color transformation sequence;

- a second reading step for reading, in the case that the color transformation sequence is to be accessed, the second pointer in the color transformation sequence to determine

whether to access the color transformation operation;

a third reading step for reading, in the case that the color transformation operation is to be accessed, the data for performing the color transformation operation from the third tag; and

a performing step for performing the color transformation operation in accordance with the data read in said third reading step to transform the input image data from one color space to another color space in place of the standardized transformation.

(28) Computer-executable process steps according to Claim 27, wherein, in a case that the color transformation operation comprises a predetermined color transformation operation, the data for performing the color transformation operation comprises third pointer data for pointing to a fourth tag, the fourth tag being at a fourth level in the hierarchy of tags.

(29) Computer-executable process steps according to Claim 28, said computer-executable process steps further comprising a fourth storing step for storing table data in the fourth tag; and

a second performing step for performing the color transformation operation defined in the third tag in accordance with the table data stored in the fourth tag to transform the input image data from one color space to another color space in place of the standardized transformation.

(30) Computer-executable process steps according to Claim 27, wherein said determining step determines whether to access the first tag by reading CMM data from a header stored in the profile format and by searching the first tag for data

corresponding to the CMM data.

(31) Computer-executable process steps according to Claim 27, wherein said performing step determines which color transformation operation to perform by matching the data read in said third reading step to a pre-stored list of color transformation operations.

(32) An apparatus for modifying a predetermined profile format having alterable tags and a plurality of predefined color transformation sequences, and of performing color transformation operations on color image data in accordance with a modified profile format, said apparatus comprising:

inputting means for inputting the color image data, override information and tag data, the tag data including color transformation sequences and color transformation operations, the color transformation sequences comprising a plurality of color transformation operations;

storing means for storing the tag data in an alterable tag in the profile format according to a hierarchical storage structure, such that the color transformation sequences are at a higher level in the hierarchical storage structure than the color transformation operations, each color transformation sequence including at least one pointer to a color transformation operation; and

processing means for processing the input color image data in accordance with the tag data stored in the alterable tag in the case that the override information has a predetermined value, said processing means performing the processing by (1) accessing a color transformation sequence in the alterable tag defined by the override information, (2) accessing a color

transformation operation defined by a pointer in the accessed color transformation sequence, and (3) processing the color image data in accordance with the accessed color transformation operation.

(33) A method of modifying a predetermined profile format having alterable tags and a plurality of predefined color transformation sequences, and of performing color transformation operations on color image data in accordance with a modified profile format, said method comprising:

an inputting step for inputting the color image data, override information and tag data, the tag data including color transformation sequences and color transformation operations, the color transformation sequences comprising a plurality of color transformation operations;

a storing step for storing the tag data in an alterable tag in the profile format according to a hierarchical storage structure, such that the color transformation sequences are at a higher level in the hierarchical storage structure than the color transformation operations, each color transformation sequence including at least one pointer to a color transformation operation; and

a processing step for processing the input color image data in accordance with the tag data stored in the alterable tag in the case that the override information has a predetermined value, said processing step comprising (1) accessing a color transformation sequence in the alterable tag defined by the override information, (2) accessing a color transformation operation defined by a pointer in the accessed color transformation sequence, and (3) processing the color

image data in accordance with the accessed color transformation operation.

(34) Computer-executable process steps stored in a computer-readable storage medium, said process steps for modifying a predetermined profile format having alterable tags and a plurality of predefined color transformation sequences, and of performing color transformation operations on color image data in accordance with a modified profile format, said process steps comprising:

- an inputting step for inputting the color image data, override information and tag data, the tag data including color transformation sequences and color transformation operations, the color transformation sequences comprising a plurality of color transformation operations;

- a storing step for storing the tag data in an alterable tag in the profile format according to a hierarchical storage structure, such that the color transformation sequences are at a higher level in the hierarchical storage structure than the color transformation operations, each color transformation sequence including at least one pointer to a color transformation operation; and

- a processing step for processing the input color image data in accordance with the tag data stored in the alterable tag in the case that the override information has a predetermined value, said processing step comprising (1) accessing a color transformation sequence in the alterable tag defined by the override information, (2) accessing a color transformation operation defined by a pointer in the accessed color transformation sequence, and (3) processing the color

image data in accordance with the accessed color transformation operation.

3. Detailed Explanation of the Invention

[Field of the Invention]

The present invention is a system for modifying a profile format, such as an International Color Consortium (ICC) profile format, by storing, in a private tag, an order of color transformation operations and pointers to color transformation operations in the profile format, both of which take precedence over other color transformation operations defined in the profile format.

[Description of the Related Art]

Profile formats are used to translate color image data formatted for one device color space into color image data formatted for another device color space or a device-independent color space. For example, an ICC profile format defines a series of standard color transformation operations to convert color image data formatted for an RGB (red, green and blue) color space in a color monitor into a device-independent color space, called the "profile connection space", and then into color image data formatted for a CMY (cyan, magenta and yellow) color space in a color printer.

While standardized ICC profiles achieve color compatibility between different devices in some cases, standardized ICC profiles are inflexible with respect to the numbers and types of color transformations operations which they perform and with respect to the order in which the color transformations are performed.

More specifically, ICC profiles define color transformation

operations in public tags. In general, a tag is a area of memory which stores formatted data used for performing color transformation operations, and/or pointers to other tags. Public tags, as defined by the ICC, comprise tags which contain standardized color transformation operations which perform a predetermined set of color transformations in a predetermined order. ICC public tags are not alterable.

In this regard, because ICC public tags are limited to ICC-defined color transformation operations, they are unable to perform certain color transformation operations, particularly those color transformation operations which are needed to convert between color spaces not covered by the ICC.

Additionally, because ICC public tags are not alterable, once a public tag is accessed to perform a color transformation, all of the pre-set color transformation operations stored in the public tag must be performed in the specified order. In some cases, this can result in errors in color transformation. In other cases, this is merely inefficient. For example, unnecessary color transformation operations may be performed on input image data, regardless of whether the color transformation operations are required.

Thus, there exists a need for a system of modifying ICC profile formats which permits a software developer to modify the type, number and order of color transformation operations stored in the ICC profile format.

[Summary of the Invention]

The present invention addresses the foregoing need by providing a system for modifying standardized color transformation operations in an ICC profile format using

private tags. According to the present invention, color transformation operations, in addition to standardized color transformation operations are stored and accessed via an ICC profile format. Advantageously, the present invention also permits random access of color transformation operations stored in public tags in the ICC profile format by using pointers in the private tags to access the color transformation operations.

Thus, according to one aspect, the present invention is a system (i.e., a method, an apparatus and computer-executable process steps) of performing a color transformation operation on input image data, the system using a first tag and a second tag to perform the color transformation operation in place of a predetermined series of color transformation operations stored in a public tag in a profile format. The system includes an inputting step for inputting color image data, a storing step for storing a first tag and a second tag, the first tag for storing override information for overriding the predetermined series of color transformations, and the second tag for storing color transformation operation data accessible via a hierarchical storage structure, and a determining step for determining, based on the override information in the first tag, whether to access the second tag. Also included in the system are a reading step for reading the color transformation operation data in the second tag in a case that the determining step determines to access the second tag, the reading step following pointers within the hierarchical storage structure to access the color transformation operation data in the second tag, and a processing step for performing a

color transformation operation on the input image data in accordance with the color transformation operation data read in the reading step.

By virtue of the foregoing configuration, it is possible to convert color image data from a first device-dependent color space, such as monitor RGB color image data, into a device-independent color space using a customized color transformation process, while still taking advantage of the features provided in standard profile format. By using the foregoing system in two different devices, it is possible to convert data from a first color space of a first device to a device-independent color space, and then from the device-independent color space to a second color space of a second device.

Additionally, because the foregoing system stores color transformation operations and calls thereto hierarchically, the foregoing system provides color matching software with ready access to the customized color transformation operations.

According to another aspect, the present invention is a system for using tags to modify a standardized transformation in a profile format in which numeric information is stored for use by the standardized transformation, and in which the tags are hierarchically stored. The system includes an inputting step for inputting color image data, and a first storing step for storing first pointer data at predetermined locations in a first tag which is at a highest level in a hierarchy of tags, the first pointer data pointing to a color transformation sequence in a second tag, the second tag being at a second highest level in the hierarchy of tags. Also included in the

system are a second storing step for storing the color transformation sequence in the second tag, the color transformation sequence including a second pointer for pointing to a color transformation operation stored in a third tag, the third tag being at a third highest level in the hierarchy of tags, and a third storing step for storing data for performing the color transformation operation in the third tag. A determining step determines whether to access the first tag based on information stored in the profile format, and a first reading step reads, in the case that the first tag is to be accessed, the first pointer information in the first tag and determines whether to access the color transformation sequence. A second reading step reads, in the case that the color transformation sequence is to be accessed, the second pointer in the color transformation sequence to determine whether to access the color transformation operation, and a third reading step reads, in the case that the color transformation operation is to be accessed, the data for performing the color transformation operation from the third tag. A performing step performs the color transformation operation in accordance with the data read in the third reading step to transform the input image data from one color space to another color space in place of the standardized transformation.

Advantageously, the foregoing aspect of the present invention permits a user selectively to create and modify a private tag which can be used to modify an overall color transformation operation performed by a profile format, and thereafter to perform the modified color transformation

operation on color image data. In addition, the foregoing hierarchical storage structure facilitates access to color transformation operation data used in the creation and modification of the private tag.

According to still another aspect, the present invention is a system for modifying a predetermined profile format having alterable tags and a plurality of predefined color transformation sequences, and of performing color transformation operations on color image data in accordance with a modified profile format. The system includes an inputting step for inputting the color image data, override information and tag data, the tag data including color transformation sequences and color transformation operations, the color transformation sequences comprising a plurality of color transformation operations. A storing step stores the tag data in an alterable tag in the profile format according to a hierarchical storage structure, such that the color transformation sequences are at a higher level in the hierarchical storage structure than the color transformation operations, each color transformation sequence including at least one pointer to a color transformation operation. A processing step processes the input color image data in accordance with the tag data stored in the alterable tag in the case that the override information has a predetermined value. The processing step includes (1) accessing a color transformation sequence in the alterable tag defined by the override information, (2) accessing a color transformation operation defined by a pointer in the accessed color transformation sequence, and (3) processing the color image

data in accordance with the accessed color transformation operation.

This brief summary has been provided so that the nature of the invention may be understood quickly. A more complete understanding of the invention can be obtained by reference to the following detailed description of the preferred embodiment in connection with the attached drawings.

[Detailed Description of the Preferred Embodiment]

Figure 1 is a view showing the outward appearance of a representative embodiment of the present invention. Shown in Figure 1 is computing equipment 1, such as a Macintosh or an IBM PC-compatible computer having a windowing environment, such as Microsoft(r) Windows. Provided with computing equipment 1 is display screen 2, such as a color monitor, keyboard 4 for entering text data and programmer commands, pointing device 6, such as a mouse, for pointing and for manipulating objects displayed on display screen 2, and printer 16 for outputting color images generated by computing equipment 1.

Computing equipment 1 includes a mass storage device such as computer disk 7, also shown in Figure 2, for storing profile formats such as ICC profile 8 which includes public tags and private tags, DOS(r) operating system, and a windowing operating system, such as Microsoft Windows(r). Also stored in computer disk 7 are Canon(r) color matching software (hereinafter "CMM") 9, private tag creation and modification application program 10, and profile manager routine 12, all of which contain stored program instructions by which computing equipment 1 manipulates and stores data

files on disk 7 and presents data in those files to an operator via display screen 2. These programs are described in more detail below.

Computing equipment 1 also includes floppy disk drive interface 14, into which floppy disks can be inserted. Information from such floppy disks can be downloaded to computer disk 7. Such information can include data files and application programs, such as CMM 9, private tag creation and modification application program 10, and profile manager routine 12. Computing equipment 1 can also include a CD-ROM interface (not shown), from which information can also be downloaded to disk 7.

Color image data is input by scanner 15 which scans documents or other images and provides bit map images of those documents to computing equipment 1. Color image data may also be input into computing equipment 1 from a variety of other sources such as network interface 17 or from other external devices via facsimile/modem interface 18.

Additionally, it should be noted that ICC profile 8 can also be accessed by computing equipment 1 from a variety of other sources such as network interface 17 or from other external devices via facsimile/modem interface 18.

It should be understood that, although a programmable general-purpose computer arrangement is shown in Figure 1, a dedicated or stand-alone computer or other type of data processing equipment can be used in the practice of the present invention.

Figure 2 is a detailed block diagram showing the internal construction of computing equipment 1. As shown in Figure 2,

computing equipment 1 includes a central processing unit (hereinafter "CPU") 20 interfaced with computer bus 21. Also interfaced with computer bus 21 is scanner interface 22, network interface 17, fax/modem interface 16, display interface 23, keyboard interface 25, mouse interface 29, main memory 30, disk 7, floppy disk drive interface 14, and printer interface 24.

Main memory 30 interfaces with computer bus 21 so as to provide random access memory storage for use by CPU 20 when executing stored program instructions such as Microsoft Windows(r). CMM 9, private tag creation and modification application program 10, profile manager routine 12, and other application programs (not shown). More specifically, CPU 20 loads those programs from disk 7, or alternatively, from a floppy disk in floppy disk drive interface 14, into main memory 30 and executes those stored programs out of main memory 30.

The present invention will be described with respect to a color image processing system which includes CMM 9. In brief, CMM 9 inputs color image data in a first color format, such as RGB and converts the color image data into a second format, such as CMY. To perform the foregoing color transformation operations, CMM 9 uses data stored in ICC profiles, which are described in "International Color Consortium Profile Format", version 3.01, (revised May 8, 1995), the contents of which are hereby incorporated by reference into the subject application.

Briefly, ICC profiles are device profiles which can be used in pairs to translate color data created on one device into a native color space of another device. For example, as

described above, RGB color image data for a monitor can be converted into CMY image data for a printer using ICC profiles.

An ICC profile provides CMM 9 with color transformation information for a particular device. More specifically, an ICC profile is provided for each device and is used by CMM 9 to transform color image data from a device-dependent color space to the profile connection space, and to transform color image data from the profile connection space to another device-dependent color space. This relationship is illustrated in Figure 3.

More specifically, Figure 3 shows monitor 31 having ICC profile 32 which is used to transform RGB image data in monitor 31 to device-independent image data in profile connection space 34. Printer 35 includes ICC profile 37 which converts the color image data from the device-independent profile connection space to CMY image data, which can be used by printer 35. In this manner, CMM 9 uses the ICC profiles to convert between two device-dependent color spaces. It should be noted that while Figure 3 shows ICC profiles 32 and 37 with respect to monitor 31 and printer 35, respectively, these ICC profiles are not resident on their respective devices. These ICC profiles may be embedded in data to be converted in accordance with the ICC profile or be stored in a memory in a connected personal computer. For example, the ICC profiles could be stored in a single memory, accessible by a single CPU.

Additionally, it is noted that ICC profiles can be used with devices in addition to a printer and a monitor, such as scanners, facsimile machines, etc.

The profile connection space is defined by a standard

illuminant of D50, the 1931 CIE standard observer, and 0/45 or 45/0 reflectance measurement geometry. The reference viewing condition is ANSI PH2.30-1989, which is a D50 arts viewing environment.

An ICC profile, an example of which is shown in Figure 4, includes two basic elements: header 39 and tag table 40. Header 39 includes information which is used by CMM 9 to process input image data in accordance with the ICC profile. The header data must be in big-endian notation. Tag table 40, which is described in more detail below with respect to Figure 4, is used by CMM 9 to access color transformation operations and other information via public and private tags.

Included in an ICC profile are required public tags, which are designed to provide a complete set of information necessary for a CMM to transform color information between the profile connection space and device-dependent color spaces. Additionally, an ICC profile can include optional public tags which can be used to perform additional transformations, and private tags, which individual developers can customize to add proprietary value to their ICC profile.

For an input device, such as a scanner, CMM 9 requires that a profile have the following tags, in order to perform a color transformation: a profile description tag, a device manufacturer tag, a device model name tag, a media XYZ white point tag, a UCCMS private information tag, a copyright tag, a red colorant tag which contains relative XYZ tristimulus values of a red channel, a blue colorant tag which contains relative XYZ tristimulus values of a blue channel, a green colorant tag which contains relative XYZ tristimulus values of

a green channel, a red channel tone reproduction curve tag, a green channel tone reproduction curve tag and a blue channel tone reproduction curve tag. Optionally, the profile can contain an AtoBn tag which defines an 8 bit or 16 bit LUT.

For a display device, such as a monitor, CMM 9 requires that a profile have the following tags, in order to perform a color transformation: a profile description tag, a device manufacturer tag, a device model name tag, a media XYZ white point tag, a copyright tag, a red colorant tag which contains relative values of red phosphor, a blue colorant tag which contains relative values of blue phosphor, a green colorant tag which contains relative values of green phosphor, a red channel tone reproduction curve tag, a green channel tone reproduction curve tag and a blue channel tone reproduction curve tag. Optionally, the profile can contain a UCCMS private information tag.

For an output device, such as a printer, CMM 9 requires that a profile have the following tags, in order to perform a color transformation: a profile description tag, a device manufacturer tag, a device model name tag, an AtoB0 Tag, a BtoA0 tag, a gamut tag, an AtoB1 tag, a BtoA1 tag, an AtoB2 tag, a BtoA2 tag, a UCCMS private information tag, an XYZ media white point tag, a measurement tag and a copyright tag. The foregoing AtoBn tags have either ICC lut8type or lut16Type structures. The general model for the ICC lut8type or lut16Type structures is:

matrix-->one-dimensional LUT-->

multidimensional LUT-->one dimensional LUT.

In an lut8type tag, input, output and color LUTs are

arrays of 8 bit unsigned values. Each input table consists of one byte integers. Also, each input table entry is appropriately normalized to a range of 0 to 255. When stored in a tag, a one-dimensional LUT is assumed to be packed one after another in ascending order according to ICC specifications.

The AtoB0 tag is used for photographic rendering. The AtoB0 tag defines a three-by-three matrix, in which matrix elements are stored in bytes 12 to 45 of the AtoB0 tag, an input channel for C, M, Y inputs is stored in byte 8, an output channel for L, a, b outputs is stored in byte 9, LUT grid points (e.g., 33x33x33) are stored in byte 10, padding is stored in byte 11, input tables (identity) are stored in bytes 46 et seq., where color LUTs and output tables are also stored for photographic rendering. Bytes 0 to 3 define the tag.

The BtoA0 tag has the same format as the AtoB0 tag except that the input tables and output tables are switched and the color LUT has the inverse of the AtoB0 tag.

The gamut tag has the same format as the AtoB0 tag except the gamut tag uses L, a, b values for the input channels and a bitmap output for the output channels.

The AtoB1 tag is used for relative colorimetric rendering and has a format that corresponds to the AtoB0 tag, while the BtoA1 tag is used for relative colorimetric rendering and has a format that corresponds to the BtoA0 tag. Both are either lut16type or lut8type.

The AtoB2 tag is used for saturation rendering and has a format that corresponds to the AtoB0 tag, while the BtoA2 tag is used for saturation rendering and has a format that

corresponds to the BioA0 tag. Both are either Int16type or Int8type.

CMM 9 also supports an optional preview profile public tag. The tag has a format identical to the AtoB0 tag except that L, a, and b data is input to the input channel.

The present invention, as described in more detail below, uses customized private tags to modify and manipulate color transformation operations performed within a predetermined ICC profile.

In the present invention, CMM 9 includes profile manager routine 12 (e.g., ColorGear(R), produced by Canon Information Systems, Inc.), which accesses ICC profile 8, and/or various public and private tags stored therein, to create or modify the private tags (and/or the device profile).

Creation or modification of a private tag is performed in accordance with the flow diagram shown in Figure 5, which depicts operation of private tag creation and modification application program 10 for entering information into private tags. More specifically, upon execution of application program 10, in step s501, it is determined whether a private tag is being created or modified. If a private tag is being created, processing proceeds to step s502, where a user is prompted on display screen 2 for information necessary to create a private tag. If the user fails to enter all of the required information, a private tag cannot be created. Examples of screens with which the user is prompted are shown in Figures 6 and 7. It is noted that screens 43 and 44, shown in Figures 6 and 7, respectively, correspond to the private tag depicted in Figure 8, which is discussed in more detail

below.

If, however, in step s501, a private tag is being modified, processing proceeds to step s506 in which the user clicks on "Get" button 42 on input screen 43 (or input screen 44). In response, application program 10 requests private tag information from profile manager routine 12 based on information on the screen. Profile manager routine 12 then accesses the required areas of memory for the private tag data and provides the requested data to application program 10.

Upon receipt of the private tag data, in step s502, application program 10 displays the information to the user on screens 43 and 44. The user can then set information in the private tag as desired by entering the information on screens 43 and/or 44 and by clicking on "Set" button 46. The user can page through private tag data using "Next" button 47 and "Previous" button 48, shown in Figures 6 and 7.

Upon entry of information onto the screen in step s503, processing proceeds to step s504, in which the user sets the private tag data by clicking on "Set" button 46. Once the user sets the private tag data, the private tag data is transmitted from application 10 to profile manager routine 12. Upon receipt of the private tag data, in step s505, profile manager routine 12 stores the private tag data in accordance with the byte assignments provided in the private tag. For example, in the ucml private tag shown in Figure 8 and described in more detail below, "Canon Signature" is stored in bytes 4 to 7, while "Creator Division" is stored in bytes 112 to 175, and so on. In this manner, a user can alter information stored within a private tag.

As described above, Figure 4 shows a screen printout of an example of ICC profile 8 which includes header 39 and tag table 40.

Header 39 provides a set of parameters specific to ICC profile 8, and is preferably stored in the first 128 bytes thereof. Parameters included in header 39 are described below.

- (1) "Size", stored in bytes 0 to 3 of header 39, defines the profile's size.
- (2) "CMType", stored in bytes 4 to 7, defines the CMM with which the profile is associated. For Canon(r) devices, this value is "UCCM".
- (3) "Version", stored in bytes 8 to 11, defines the profile's version number. This is defined by the ICC as 2000000H.
- (4) "ProfileClass", stored in bytes 12 to 15, defines the profile's class, and may be any one of "prtr" (printer), "mnr" (monitor), "scnr" (scanner), "link" (link device), "spac" (color space conversion), and "abst" (abstract profiles).
- (5) "DataColorSpace", stored in bytes 16 to 19, defines the color format into which the profile converts color image data, and may be any one of "RGB", "XYZ", "GRAY", "CMY", "Luv", "HSV", "CMYK", "YCbCr", "HLS", "Lab" and "Yxy".
- (6) "InterchangeSpace", stored in bytes 20 to 23, defines the profile connection space, and may be either "LAB" or "XYZ".
- (7) "CreationDate", stored in bytes 24 to 35, defines the date and time that the profile was created.
- (8) "CS2Signature", stored in bytes 36 to 39, defines the profile's file signature, which is used by the operating system of a device using the profile to create an icon.

This value is set to "acsp" for all ICC profiles.

(9) "Prim.platform", stored in bytes 40 to 43, defines the primary platform or operating system for which the profile was created, and may be any one of the following values:

"Appl" (Apple operating system), "MSFT" (Microsoft operating system), "SGI" (Silicon Graphics), "SUNW" (Sun) and "TGNT" (Taligent).

(10) "Flags", stored in bytes 44 to 47, defines various hints for a CMM, such as distributed processing and caching options. This parameter is not used in Canon(r) devices and is therefore set to 0H.

(11) "DeviceManufacturer", stored in bytes 48 to 51, is the signature for the manufacturer of the device with which the profile is to be used. This parameter has a value of "CANO" for Canon(r) devices.

(12) "DeviceModel", stored in bytes 52 to 55 defines the model number or name of the device with which the profile is to be used. The value of "DeviceModel" must be conform to the standard imposed by the ICC, Apple ColorSync(R) and Microsoft ICM(r). Specifically, the model number or name must be a 4 byte ASCII string using characters "A" to "Z" (capital letters only and characters "0" to "9"). The format of "DeviceModel" should be:

<1st byte: Division ID>

<2nd-4th byte: Model number and extension>.

where "Division ID" identifies the company or division which created the product. Ideally, each company or division should have a unique "Division ID". In Canon Inc., divisions are defined as follows: "B" stands for bubble jet, "C" stands for

copier, "D" stands for digital camera, "F" stands for facsimile, "L" stands for laser beam printer, "M" stands for Canon(r) monitor, "S" stands for scanner, "V" stands for video camera, "Z" stands for third-party, non-Canon(r) product files. These values are randomly determined and can be set as desired. In preferred embodiments, bytes 2-4 of "deviceModel" are encoded according to one of the following two formats.

a. Format 1:

<1 byte: Division byte>

<2 bytes: model name>

<1 byte: extension>

As described above, the division byte stores the division of the device using the profile. The model name bytes store the device model. The extension byte stores the extension of the device, e.g., BJ-600e would be an extension of a BJ-600 device. To avoid two devices having a same "deviceModel", the following convention has been adopted. A-I are used as extension indicators for nine extensions; J-R are used as the next nine extension indicators when there is a collision of the first three bytes; S-Z are used as extension indicators for the next 8 extensions; and 0-9 are used as extension indicators for the next 10 extensions. An example of this system is shown below:

BJC-4000 ---> B40A (first extension of BJC 4000).

BJC-4000E --> B40B (second extension of BJC-4000),

BJC-400 ---> B40J (first extension of BJC-400), and

BJC-400X ---> B40K (second extension of BJC-400).

b. Format 2:

<1 byte: Division byte>

<3 bytes: Model ID + extension>

In this format, division byte is the same as division byte described above, and model ID and extension number are determined so as to ensure that two different devices do not have the same "deviceModel". In the second format, a mathematical algorithm is used to calculate a unique model ID + extension from the actual model number, e.g., from the "600" in BJC-600. However, any method may be used so long as it produces unique "deviceModel" values for each device.

(13) "DeviceAttributes", stored in bytes 56 to 63, are attributes which are unique to the particular device setup. In the present invention, media type, resolution, halftoning ink type and creator must be specified. A preferred format for "deviceAttributes" is as follows: bytes 56 to 57 reserved for the ICC, byte 58 for media type, byte 59 for resolution, byte 60 for screening, byte 61 for ink type and bytes 62 to 63 reserved for future use.

In a preferred embodiment, media type, resolution, screening and ink type have values defined as follows:

a. Media Type

- 0: Not Applicable/Do Not Care
- 1: Plain Paper
- 2: Coated Paper
- 3: Glossy Paper
- 4: OHP Paper
- 5: High Glossy Film
- 6: Fine Coated Paper
- 7: BPF Paper
- 8: Textile Paper

b. Resolution

- 0: Not Applicable/Do Not Care
- 1: 180x180 dots per inch (dpi)
- 2: 200x200 dpi
- 3: 300x300 dpi
- 4: 360x360 dpi
- 5: 400x400 dpi
- 6: 600x300 dpi
- 7: 600x600 dpi
- 8: 720x360 dpi
- 9: 720x720 dpi
- 10: 1200x600 dpi
- 11: 1200x1200 dpi

c. Screening

- 0: Not Applicable/Do Not Care
- 1: Pattern 1
- 2: Pattern 2
- 3: Pattern 3
- 4: Erroneous Diffusion
- 5: Continuous tone Pattern 1
- 6: Continuous tone Pattern 2
- 7: Continuous tone Pattern 3

d. Ink Type

- 0: Not Applicable/Do Not Care
- 1: normal ink
- 2: ink type 1

Modifications to the foregoing values can be performed as required.

(14) "RenderingIntent", stored in bytes 64 to 67, defines an

intent (i.e., perceptual, relative colorimetric, saturation and absolute) for which the profile is designed. In the present invention, this value is 0.

(15) "WhiteXYZ", stored in bytes 68 to 79, defines illuminant values of the profile connection space. In the present invention, this value is set to D50.

(16) Bytes 80 to 127 are reserved by the ICC for future use. Header 39 is not intended to be an exhaustive list of all information which can be stored in a profile header. Rather, it is merely shows examples of information that can be stored in a profile header.

Other information, in addition to that described above, may be added to the header as needed or desired. To this end, the ICC has reserved 48 bytes, as indicated above, for future use. However, in order to implement the present invention, all of the information shown in header 39 must be present.

Tag table 40 includes a list of tags, both public and private, and information relating to the tags. In the profile, tag table 40 begins at byte 128 (i.e., after header 39). More specifically, as shown in tag table 40, "Ind" indicates the number of tags in tag table 40, "Sigmat" is a tag signature, and is used by CMY 9 to locate a particular tag; "elementOffset" is the location in memory, in both decimal and hexadecimal numerals, at which the tag begins; and "size" defines the size of the tag in bytes.

Each individual tag in tag table 40 is structured such that bytes 0 to 3 thereof define the tag signature, bytes 4 to 7 define an offset value which points to the beginning of the tag data, and bytes 8 to 11 defines the number of bytes in the

tag.

As stated above, public tags define stock color transformation operations which are available for use in all ICC profiles. An example of a public tag provided in tag table 40 is A2B2, which contains a 3x3 color transformation matrix process, a three-dimensional LUT and two sets of three one-dimensional LUTs. Other examples of public tags are described in more detail in the International Color Consortium Profile Format document.

In tag table 40, tags having "Ind" values of 0 to 16 comprise public tags, while tags having "Ind" values of 17 and 18 comprise private tags. In this case, the private tags are Canon(R)-registered private tags.

The present invention will be described with respect to the foregoing two private tags, in conjunction with the public tags listed in tag table 40. It should be noted, however, that the present invention can be used with any of the public tags listed in the International Color Consortium Profile Format document or with any tags compatible therewith.

In brief, the present invention is a system of performing a color transformation operation on input image data, the system using a first tag and a second tag to perform the color transformation operation in place of a predetermined series of color transformation operations stored in a public tag in a profile format. The system includes an inputting step for inputting color image data, a storing step for storing a first tag and a second tag, the first tag for storing override information for overriding the predetermined series of color transformations, and the second tag for storing color

transformation operation data accessible via a hierarchical storage structure, and a determining step for determining, based on the override information in the first tag, whether to access the second tag. Also included in the system are a reading step for reading the color transformation operation data in the second tag in a case that the determining step determines to access the second tag, the reading step following pointers within the hierarchical storage structure to access the color transformation operation data in the second tag, and a processing step for performing a color transformation operation on the input image data in accordance with the color transformation operation data read in the reading step.

The present invention stores color transformation operations and calls to other public and private tags in a hierarchical structure. This hierarchical structure is shown in Figure 9, which will be discussed in detail below in connection with a detailed discussion of each of the elements shown therein.

The private tag having an "Ind" value of 17, hereinafter referred to by its signature "ucml" and reference numeral 52, contains information concerning the interaction of private tags in ICC profile 8 with public tags in ICC profile 8. In addition, ucml 52 contains information required by CMM 9 to perform color matching using the private tags. If ucml 52 is missing, CMM 9 uses default settings to interpret the profile. As shown in Figure 8, this information is stored byte-sequentially in ucml 52.

More specifically, column 53 of ucml 52 defines bytes of

ucml 52 which store the data in column 56. Column 54 describes the data stored in the bytes of column 53, and column 57 defines the format for the data stored in the bytes of column 53. At this point it should be noted that "uint32" stands for a 32-bit unsigned integer, while the notation "0x" followed by a number indicates that the number is in hexadecimal format.

The data stored in ucml 52 includes "ucmsPrivateInformationTag Signature", which is a tag-specific identifier, by which CMM 9 can access ucml 52; "Canon Signature", which is a secondary identifier, for identifying the private tag as a Canon(R) private tag; "Size of parameters in bytes", which define parameter data sizes for parameters (to be described in more detail below) used in color transformation operations; "Engine version", which defines the minimum CMM version required to read the profile; "Profile Format Document version", which defines the version of the profile; "Profile version", which defines the profile version assigned by its creator; "Profile Build number", which defines the build number of the subject profile; "Interpolation flag", which defines the type of linear interpolation to be used by the profile; override elements, stored in bytes 32 to 71 (discussed in greater detail below), which store the Sequence Canon ID number required to override a particular public tag, such as A2B2; optimization flag values, stored in bytes 72 to 111, which store optimization flags for CMM 9 which indicate printing mode type, e.g., draft mode; "Creator Division", which defines the Canon(R) division that created the profile; "Support Division", which defines the Canon(R) division that

supports the profile; Von Kries Flag, which determines whether or not to use a Von Kries color transformation; and reserved bytes.

Figure 10 is a flow diagram which shows the process by which CMM 9 navigates through ICC profile 8 and the public and private tags associated therewith. More specifically, in step s1001, CMM 9 reads the CMType of profile 8 found in header 39. CMType corresponds to a pointer to an area of memory which contains the value stored therein.

Next, in step s1002 of Figure 10, if CMM 9 determines, based on the CMType in header 39, that profile 8 is a "UCCM" profile, as is the case for profile 8, processing proceeds to step s1003 in which CMM 9 performs color transformation operations in accordance with ucml 52, and not in accordance with public tags. Thus, as shown in Figure 9, ucml 52 is at the top of the storage hierarchy.

More specifically, CMM 9 reads ucml 52's signature found in bytes 0 to 3 of ucml 52, and accesses the "elementOffset" area in memory that corresponds to ucml 52, in this case byte 25667810 (see Figure 4).

Thereafter, in step s1003 of Figure 10, CMM 9 reads the data in bytes 32 to 71 of ucml 52, and based thereon determines whether any of the operations defined by the public tags in profile 8 are overridden. In this regard, if there is a 0H in an override area of ucml 52, then CMM 9 performs color transformation according to a process stored in a corresponding public tag. If there is a non-zero value in an override area of ucml 52, then CMM 9 performs color transformation according to the process stored in ucml 52 and

defined by the non-zero value. For example, as shown in Figure 8, AtoB0Tag override 59 has a value of "0x0" or 0H. Based on this information, CMM 9 will perform any AtoB0Tag color transformation process in accordance with color transformation processes stored in the AtoB0Tag public tag. Accordingly, with respect to the AtoB0Tag color transformation process, flow proceeds to end in Figure 10.

However, as also shown in Figure 8, BtoA2Tag override 60 has a value of "0x33" or 33H, which serves as a pointer to an area in memory. Therefore, with respect to color transformation operations for BtoA2Tag, processing proceeds in accordance with information stored in ucml 52, and not in accordance with the BtoA2Tag public tag.

More specifically, as shown in Figure 10, processing proceeds to step s1005, in which the private tag having "Ind" value of 18, hereinafter referred to by its signature "ucmP" and reference numeral 62, is automatically read by the CMM.

Figure 11 shows an example of ucmP 62. As shown in Figure 11, ucmP 62 stores information byte-sequentially in the same format as ucml 52. For the sake of brevity, a detailed description of the storage format is therefore omitted.

The information stored in ucmP 62 comprises "ucmcsPrivateSequencesTagTable Signature", which is a tag-specific identifier, by which CMM 9 accesses ucmP 62; "Canon Signature", which is identical to "Canon Signature" described above with respect to ucml 52; "Number of Sequences", which defines the number of color transformation sequences to be performed via ucmP 62; and "Sequences structures", which define a series of color transformation operations to be

performed in the particular color transformation sequence, along with other relevant information required by CMM 9 to perform the color transformation sequence. It is noted that while Figure 11 shows merely four sequences structures, any number of sequences structures can be added to ucmP 62, the number of sequences structures being limited only by the hardware on which the present invention is run.

The sequences structure for a color transformation sequence is shown in Figure 12. It is noted that while particular operations in the color sequences can vary, along with other parameters therein, the structure of the sequences is fixed.

More specifically, all color transformation sequences contain "CanonID", which is the sequence by which CMM 9 accesses the particular sequences structure. In this case, the "CanonID" is 33H, which corresponds to BtoA2Tag override 60 from ucmI 52 in Figure 8. Through this correspondence, CMM 9 is able to determine, based on ucmI 52, which color matching sequence to access in ucmP 62.

The color transformation sequences structure also includes "reserved bytes", which for the time being maintains a value of 00H; "numOps", which define the number of operations in the color transformation sequence, and which is discussed in more detail below; "seqPCS", which defines the profile connection space into which the subject profile converts data; "length", which defines the length in bytes of the parameters used in the operations to be performed by the color transformation sequence; and "offset", which defines the beginning point in memory of the color transformation

operations to be performed by the color transformation sequence.

As shown in Figure 12, the sequences structure also defines the format of the data stored therein. In this case, as shown, the format is "uint32", which, as indicated above, corresponds to a 32 bit unsigned integer.

The sequences structure comprises a byte string, which is stored in column 65 of ucMP 62 in bytes 16 to 39, shown in Figure 11. Thus, for the example shown in Figures 11 and 12, "CanonID" is 33, "reserved" is 00, "numOps" is 04, "seqPCS" is LAB, "length" is 40, and "offset" is 5163. These values are stored as data in bytes 16 to 39 of sequences structure 1. Similarly identifiable sequences structures can also be stored in sequences structures 2 to 4. However, for the sake of brevity, the present invention will only be described with respect to the foregoing sequences structure. Accordingly, as shown in Figure 9, ucMP 62 is next in the hierarchical storage structure after ucml 52, followed by various available sequences structures 66 to 68.

To summarize up to this point, since a 33H was present in BtoA2Tag override 60 in ucml 52, and since sequences structure 1 has a CanonID of 33H, CMM 9 reads sequences structure 1.

As described above, each sequences structure includes a "numOps" value, a "length" value and an "offset" value. CMM 9 uses these values to access color transformation operations stored within the sequences structures. More specifically, since (1) each color transformation operation is of a fixed length, defined by "length", (2) the number of color transformation operations, defined by "numOps", is

known, and (3) the area in memory where the color transformation operations are sequentially stored is known, CMM 9 can parse each color transformation operation and perform the operation based on information stored in the sequences structure. This process will be described in more detail below.

Figure 13 shows the structure of a color transformation operation. More specifically, the structure of color transformation operations is defined as follows: "oper" defines type of color transformation operation to be performed; "subid" defines a subidentification number for a particular color transformation operation, which is described in more detail below; "length" defines the length of a parameter list in the color transformation operation; "tagFlag" defines a call to another public or private tag in the case that "tagFlag" has a non-zero value, or a call to color transformation operations stored within the parameter list in the case that "tagFlag" has a zero value; and "parm" defines the parameter list used in the color transformation operation.

There are seven color transformation operations that can currently be performed using the present invention. These seven color transformation operations, each of which is defined by a UCCMS Operation Number in parentheses, include (1) NxM matrix, (2) three one-dimensional LUTs, (3) three-dimensional LUT, (4) ColorSpace transformation, (5) Shift Hues (color warping), (6) Business Graphics transformation, and (7) one three-dimensional tri-linear or pyramidal LUT. Color transformation operations can be added to or deleted from this

list as desired.

Each of the foregoing color transformation operations performs different operations on input color image data. For example, the $N \times M$ matrix could be a 3×3 matrix which is used to transform color image data from an XYZ color space to an RGB color space. Similar color transformations could be performed in accordance with other operations accessed from a sequences structure.

CMM 9 stores a list of the foregoing seven operations. When CMM 9 determines, based on the "opers" value in a sequences structure, that one of the operations is to be performed on color image data, CMM 9 reads the data for the operation to be performed from a private tag which specifies the color transformation operation, and performs the color transformation operation on the color image using the data. The following describes accessing data for the operation to be performed from a specified private tag.

More specifically, in accordance with Figure 10, each color transformation operation structure is read from the sequences structure, and based thereon, it is determined which type of color transformation operation to perform. For example, if CMM 9 reads a "1" as the "oper" value, which acts as a pointer to an area in memory, for a particular color transformation operation, CMM 9 accesses the color transformation operation that corresponds to "1", which in this case is $N \times M$ matrix. Figure 9 further shows the relationship of color transformation operations to the sequences structure 66, i.e., they are lower in the hierarchy than sequences structure 66.

Figure 14 shows an example of NxM tag 80, which contains data used to perform UCCMS Operation 1, the NxM matrix. As shown in Figure 14, the format of the private tag used to implement the NxM matrix is similar to that used to implement ucmP 62. Accordingly, a detailed description of the format will be omitted for the sake of brevity. It is noted that the format and structure of each of the private tags used for color transformation operations is the same as that of NxM tag 80.

NxM tag 80, shown in Figure 14, includes "Operation ID", which as described above, defines the type of color transformation operation performed by the private tag; "Sub ID", which is described in more detail below, and which is not used in this case; "Length of Parameter List in bytes", which defines the length of the parameter list described below, and which is used to access the parameter list; Tag Flag, which is not used in this case, and which is described in more detail below; and "Parameter Data", which defines parameters to be used in the color transformation operation. As shown in Figure 14, the parameters used in the NxM matrix, in this case N and M both being 3, are the N dimension of the matrix, the M dimension of the matrix, and each of the matrix elements.

Thus, when CMM 9 determines, in accordance with step s1006 of the flow diagram shown in Figure 10, that an NxM color transformation operation is to be performed, CMM 9 reads the parameter data stored in NxM tag 80, and performs the color transformation operation, defined by "Operation ID", in accordance with the read parameters in step s1007.

In the case that an "oper" value for a color

transformation operation defined in sequences structure 1 is 2, processing proceeds to step s1008, and thereafter to step s1009. Steps s1008 and s1009 define a color transformation operation which includes a one-dimensional LUT. Figure 15, comprised of Figures 15A and 15B, shows two examples of private tags which could be used to create LUTs for color transformation operation 2.

Figure 15A shows 1DLUT tag 81 that defines a LUT for color transformation operation 2, in which a call is made to a public tag or another private tag from 1DLUT 81. More specifically, in Figure 15A, the "Tag Flag" value is "A2B2". "Tag Flag" defines a call to another tag, in this case the "AtoB2" public tag. Thus, when CMM 9 reads 1DLUT 81 and encounters "A2B2" as the value for "Tag Flag", CMM 9 accesses the "AtoB2" public tag and performs color transformation operations in accordance therewith. In this case, therefore, no "Parameter Data" is required, since the color transformation operation is performed in accordance with data stored in a public tag. Consequently, the "Length of Parameter List" also has a value of zero.

Advantageously, the present invention permits a user, by setting a "Sub ID" value, to select a particular color transformation operation within the public tag specified in "Tag Flag". That is, each public tag, such as "AtoB2", may contain more than one color transformation operation. The present invention provides for selection of one of the color transformation operations within the public tag. For example, public tag "AtoB2" contains a 3x3 matrix and 3 LUTs: an input LUT from lut8type/lut16type, an output LUT from

lut8type/lut16type and a LUT from curve-type. By setting the "Sub ID" value to 0, 1 or 2 in 1DLUT tag 81, CMM 9 will only perform one of the color matching operations provided in public tag "AtoB2". Moreover, since CMM 9 knows, based on the "Operation ID" in 1DLUT tag 81, that LUT operations are to be performed, CMM 9 will only select a LUT stored in public tag "AtoB2", and will not select, for example, the 3x3 matrix.

In a preferred embodiment of the present invention, a value of "0" in "Sub ID" directs the CMM to use the input LUT from lut8type/lut16type; a value of "1" in the "Sub ID" directs CMM 9 to use the output LUT from lut8type/lut16type; and a value of "2" in the "Sub ID" directs CMM 9 to use the LUT from curve-type, all of which, as indicated above, are stored in public tag "AtoB2". Similar specifications can be made for each of the public tags shown in Figure 4, and for any other public tags which may contain plural color transformation operations.

Figure 15B shows an example of 1DLUT tag 82 for color transformation operation 2, in which "Tag Flag" is zero, i.e., in which color transformation operations are to be performed in accordance with information in the private tag, and not in accordance with information in a public tag. Thus, in 1DLUT tag 82, "Length of Parameter List" is 77110, which, as described above, indicates the length in bytes of the "Parameter Data" stored in bytes 16 et seq. of 1DLUT tag 82. The "Parameter Data" defined in 1DLUT tag 82 includes parameter data necessary to perform a color transformation operation, such as a gamma correction, using a LUT. Thus, as shown in Figure 15B, the "Parameter Data" includes the

dimensions of the LUT, the number of input and output bits for input and output data, respectively, and values for the LUT. As noted above, the number of parameters, and thus the size of the LUT, are not limited by the private tag. Rather these values are only limited by the capacity of the computing equipment on which the present invention is implemented.

Returning to Figure 10, in the case that an "oper" value for a color transformation operation defined in sequences structure 1 is 3, processing proceeds to step s1010 and thereafter to step s1011. Steps s1010 and s1011 define a three-dimensional LUT color transformation operation. Figure 16 shows an example of a private tag which implements color transformation operation 3. A description of Figure 16 is similar to that provided above with respect to Figure 15. accordingly a detailed description of Figure 16 is omitted for the sake of brevity.

Returning to Figure 10, in the case that an "oper" value for a color transformation operation defined in sequences structure 1 is 4, processing proceeds to step s1012, and thereafter to step s1013. Steps s1012 and s1013 define a color space transformation operation.

Figure 17 shows an example of a private tag (colorspace tag 84) which implements color transformation operation 4. In Figure 17, "Operation ID" is 4, "Length of Parameter List" is 0 and "Tag Flag" is 0H. These values do not change, in this case. "Sub ID", in this case, defines which type of color space transformation operation is to be performed via colorspace tag 84. The following color space transformations, defined by "Sub ID" values 1 to 9, can be performed via

colourspace tag 84:

- 0: RGB (Red, Green, Blue) ---> HLS (Hue, Lightness, Saturation),
- 1: HLS ---> RGB,
- 2: CMY (Cyan, Magenta, Yellow) ---> HLS,
- 3: HLS ---> CMY,
- 4: CIE XYZ ---> CIE LAB,
- 5: CIE LAB ---> CIE XYZ,
- 6: CIE xyY ---> CIE XYZ,
- 7: CIE XYZ ---> CIE xyY,
- 8: CIE XYZ ---> CRGB (Canon RGB), and
- 9: CRGB ---> CIE XYZ.

It should be noted that the foregoing list of color space transformations is not exhaustive, and additional color space transformation operations could be added thereto as desired.

In the case that an "oper" value for a color transformation operation defined in sequences structure 1 is 5, processing in Figure 3 proceeds to step s1014, and thereafter to step s1015. Steps s1014 and s1015 define a color warping transformation operation. Briefly, a color warping transformation operation uses input RGB hue angles to correct CMYK errors in output color image data.

Figure 18 shows an example of warping tag 88 which implements color transformation operation 5. In Figure 18, "Operation ID" is 5, "Sub ID" is 0 and "Tag Flag" is 0B. These values do not change. In this case, the "Parameter Data" includes values necessary to perform the color warping transformation function, i.e., the RGB hue angles.

In the case that an "oper" value for a color

transformation operation defined in sequences structure 1 is 6, processing in Figure 10 proceeds to step s1016, and thereafter to step s1017. Steps s1016 and s1017 define a business graphics LUT which is useful, for example, for performing color transformation operations as described in currently-pending U.S. Patent Application No. 08/496,100, entitled "Color Management System Having Business Graphics Rendering Mode", the contents of which are hereby incorporated by reference into the present application. Business tags 89 and 90, shown in Figures 19A and 19B, respectively, correspond to 1DLUT tags 81 and 82, shown in Figures 15A and 15B, respectively.

As was the case above with respect to the private tags shown in Figure 15, the private tags shown in Figure 19 are accessed by CMM 9 via sequences structure 1 in wcmP 62. However, in the case that input data to be processed in accordance with the private tags shown in Figure 11 have same values, CMM 9 does not perform a color transformation operation in accordance with either of the private tags shown in Figure 19. Instead no color transformation operation is performed. Since this is the only difference between the color transformation performed in accordance with the private tags shown in Figure 19 and those shown in Figure 15, a detailed description of the color transformation operations defined by the private tags shown in Figure 19 is omitted for the sake of brevity.

In the case that an "oper" value for a color transformation operation defined in sequences structure 1 is 7, processing in Figure 3 proceeds to step s1018, and thereafter

to step s1019. Steps s1018 and s1019 define a tri-linear or pyramidal interpolation process for performing three-dimensional linear interpolation.

In a preferred embodiment of the present invention, the private tag which performs tri-linear or pyramidal interpolation is configured similarly to the other private tags described above. Figure 20 shows a preferred embodiment of private tag 92 which is used to perform color transformation operation 7. As shown in Figure 12, in private tag 92, the "Operation ID" is 7, the "Sub ID" is 0 and the "Tag Flag" is zero. These values do not change. Also included is the "Length of Parameter List", which has a value that corresponds to the number of bytes comprising the parameters, and the "Parameter Data", which defines data for performing tri-linear or pyramidal interpolation.

Another embodiment of a private tag used for performing tri-linear or pyramidal interpolation in accordance with the present invention is shown in Figure 21. More specifically, instead of including parameter data with which to perform a requisite color transformation operation, private tag 93 shown in Figure 21 includes "Table ID". The "Table ID" value refers to another private tag which is used in the determination of whether tri-linear interpolation or pyramidal interpolation is to be performed. An example of this other private tag is shown in Figure 22.

More specifically, Figure 22 shows an example of private tag 94, called from private tag 93 shown in Figure 21, which defines tables used to perform tri-linear or pyramidal interpolation.

Thus, as shown in Figure 22, private tag 94, hereinafter referred to as mcmT 94, which is structured the same as other private tags in the present invention, includes the following elements: "accmsPrivateInterpolation TableTag Signature", which is a tag-specific identifier, by which CMM 9 can access this private tag; "Canon Signature", which was described above with respect to Figure 8; "Number of Tables", which defines the number of LUTs in mcmT 94; and "Table" N, where N = 1, 2, ..., which are LUTs, according to which CMM 9 performs tri-linear or pyramidal interpolation.

The "Table" values include a table header structure, an example of which is shown in Figure 23. As shown in Figure 23, the table header structure includes "tableID", which is a table-specific identifier; "length", which defines the length of the specified table; and "offset", which defines the point in memory at which the specified table begins. This information can be used to access different table in memory. As shown in Figure 9, these table are accessed only from color transformation operation 7.

An example of a private tag which defines a LUT that can be accessed by mcmT 94 is shown in Figure 24. LUT private tag 96 shown in Figure 24 has a format identical to other private tags used in the present invention. LUT private tag 96 includes the following elements: "tableKind", which defines the type of operation performed by the LUT, i.e., either tri-linear interpolation, in which case "tableKind" has a value of 1, or pyramidal interpolation, in which case "tableKind" has a value of 0; "parentTag", which defines the parent tag for the LUT; "inChannels", which defines a number of input channels;

"inBits", which defines a number of input bits used to address the LUT; "outChannels", which defines a number of output channels; "outBits", which defines a number of bits used for output, usually both this number and "inBits" is 8; "gridPoints3DTable", which defines the number of grid points in the LUT; and "gridPoints4DTable", which defines the number of grid points for a fourth channel, and which is used in pyramidal interpolation only. As described above, the foregoing information is used by the color matching method to perform tri-linear or pyramidal interpolation.

If parentTag is 0X00000000 and tableKind is tri-linear, then an entire interpolation color LUT is pointed to by ucMT 94. If a parentTag is 0X00000000 and tableKind is pyramidal, then a first dataset in the color LUT is defined as major grid data followed by minor grid data. If parentTag is the signature of a three-dimensional LUT in profile 8 and tableKind is tri-linear, the three-dimensional LUT is used as grid points and the rest of the fields, like inChannels are ignored. If parentTag is the signature of a three-dimensional LUT in profile 8 and tableKind is pyramidal, the three-dimensional LUT is used as major grid points and the color LUT contains only minor grid points.

Referring back to Figure 10, in the case that none of the operations of steps s1006 to s1018 is performed, processing ends.

Additionally, it is noted that a private data tag, called "PrivateDataTableTag", can be modified by a user to include a user's private data, and stored at a point within the hierarchical storage structure.

A brief explanation will now be provided, with reference to Figure 25, of transforming color image data from color monitor 2 (RGB) color space to color printer 16 (CMY) color space, in accordance with the present invention. It should be noted that while such a color transformation process could possibly be performed using currently-available public tags, the following example will be described with respect to user-created private tags, so as to illustrate the invention.

Assuming there exists predefined ICC profiles for both color monitor 2 and color printer 16, in step s2501 a user executes above-described application program 10 for creating and modifying private tags. In accordance therewith, the user enters private tag information for transforming color image data from an RGB color space to a profile connection space for color monitor 2, and private tag information for transforming color image data from the profile connection space to a CMY color space for color printer 16.

Once private tags have been created for both color monitor 2 and color printer 16 in step s2501, processing proceeds to step s2502. In step s2502, CMM 9 inputs RGB color image data from color monitor 2. Thereafter, in step s2503, CMM 9 accesses the ICC profile for transforming the color image data from the RGB color space to the profile connection space.

Once CMM 9 accesses the color profile for color monitor 2, CMM 9 reads the profile to determine which tags listed therein to access. In this case, CMM 9 would read "UCCM" under CMType, which would indicate that processing would proceed in accordance with the ucml private tag, and not in accordance with standard ICC processes.

It must be noted that the ICC profiles used in the present example have a structure identical to that described above. Accordingly, in step s2504, CMM 9 reads the ucml to determine which public and/or private tags are to act on the RGB data. In the present case, assuming that there is a non-zero value in an override area of the ucml, CMM 9 locates the same non-zero value in one of the sequences structure areas of the ucmP.

Thus, by virtue of the foregoing steps, CMM 9 has determined which sequences of color transformation operations are to be performed on the RGB color image data, i.e., the color transformation operations accessed via the sequences structure that corresponds to the non-zero value in the override area of the ucml.

Once CMM 9 determines the sequences of color transformation operations to be performed on the RGB color image data, CMM 9 accesses the data for each of the color transformation operations based on an "cper" value and storage location data for each color transformation operation in the sequences structure. In this manner, CMM 9 accesses the private tags that store color transformation operation data, such as elements in an NxM matrix.

Next, in step s2505, CMM 9 performs the color transformation operation, indicated by the "cper" value in the sequences structure, on the RGB color image data, using the parameters from the private tag referenced in the sequences structure. In the present case, the parameters and color transformation operation would be set so as to transform color image data from an RGB color image space to the profile connection space.

Following conversion to the profile connection space, processing proceeds to step s2506. In step s2506, CMM 9 accesses the ICC profile for color printer 16. It is noted that while the ICC for color printer 16 and the ICC for color monitor 2 can be stored in a single memory, such as disk 7, or in separate memories, storage in a single memory facilitates access by CMM 9.

Once CMM 9 accesses the color profile for color printer 16, CMM 9, as was the case above with respect to the ICC profile for color monitor 2, reads the profile to determine which tags listed therein to access. In this case, the CMM would read, under CMType, "UCCM", which would indicate that processing would proceed in accordance with the ucmI private tag, and not in accordance with standard ICC procedures.

Thus, as described above, in steps s2506 to s2508, CMM 9 would access the ucmI and the ucmP private tags of the ICC profile for color printer 16 in order to transform the color image data from the profile connection space to the CMY color space of color printer 16. Since an overview of this processing is identical to that described above with respect to the ICC profile for color monitor 2, a detailed description thereof will be omitted for the sake of brevity.

Following step s2508, processing proceeds to step s2509, in which the color-transformed CMY color image data is output to printer 16.

It is noted that while the present invention has been described with respect to an ICC profile format, the present invention can be used with any type of cross-platform device format in which color transformation operations can be stored

and accessed.

Likewise, it is further noted that while the present invention has been described with respect to a color monitor and a color printer, the present invention can be utilized with other color image processing devices, such as a color scanner or a color facsimile machine.

The invention has been described with respect to a particular illustrative embodiment. It is to be understood that the invention is not limited to the above-described embodiment and that various changes and modifications may be made by those of ordinary skill in the art without departing from the spirit and scope of the appended claims.

[Other Embodiment]

The present invention can be applied to a system constituted by a plurality of devices (e.g., host computer, interface, reader, printer) or to an apparatus comprising a single device (e.g., copy machine, facsimile).

Further, the object of the present invention can be also achieved by providing a storage medium storing program codes for performing the aforesaid processes to a system or an apparatus, reading the program codes with a computer (e.g., CPU, MPU) of the system or apparatus from the storage medium, then executing the program.

In this case, the program codes read from the storage medium realize the functions according to the embodiment/embodiments, and the storage medium storing the program codes constitutes the invention.

Further, the storage medium, such as a floppy disk, a hard disk, an optical disk, a magneto optical disk, CD-ROM, CD-R, a

magnetic tape, a non-volatile type memory card, and ROM can be used for providing the program codes.

Furthermore, besides aforesaid functions according to the above embodiment/embodiments are realized by executing the program codes which are read by a computer, the present invention includes a case where an Operating System (OS) or the like working on the computer performs a part or entire processes in accordance with designations of the program codes and realizes functions according to the above embodiment/embodiments.

Furthermore, the present invention also includes a case where, after the program codes read from the storage medium are written in a function expansion card which is inserted into the computer or in a memory provided in a function expansion unit which is connected to the computer, CPU or the like contained in the function expansion card or unit performs a part or entire process in accordance with designations of the program codes and realizes functions of the above embodiment/embodiments.

4. Brief Explanation of the Drawings

Figure 1 shows a perspective view of computer hardware used in an operation of the present invention.

Figure 2 shows a block diagram of the present invention.

Figure 3 shows the relationship between device-dependent color spaces and the profile connection space.

Figure 4 shows an ICC profile.

Figure 5 shows a flow diagram for creating and modifying private tags.

Figure 6 shows a screen which depicts private tag information.

Figure 7 shows a screen which depicts private tag

information.

Figure 8 shows the ucml private tag.

Figure 9 shows the hierarchical relationship between the private tags created by the present invention.

Figures 10A and 10B shows a flow diagram for setting up a color transformation process according to the present invention.

Figure 11 shows the ucMP private tag.

Figure 12 shows a sequence structure.

Figure 13 shows a color transformation operation structure.

Figure 14 shows an NxM matrix private tag.

Figures 15A and 15B shows examples of three one-dimensional look-up table private tags.

Figure 16 shows a three-dimensional look-up table private tag.

Figure 17 shows a colorspace private tag.

Figure 18 shows a color warping private tag.

Figures 19A and 19B shows examples of business graphics look-up table private tags.

Figure 20 shows a first embodiment of a tri-linear interpolation or pyramidal interpolation private tag.

Figure 21 shows a second embodiment of a tri-linear interpolation or pyramidal interpolation private tag.

Figure 22 shows a ucMT private tag.

Figure 23 shows a table structure.

Figure 24 shows a table private tag.

Figure 25 is a flow diagram showing process steps for converting input image data from one color space to another color space using the present invention.

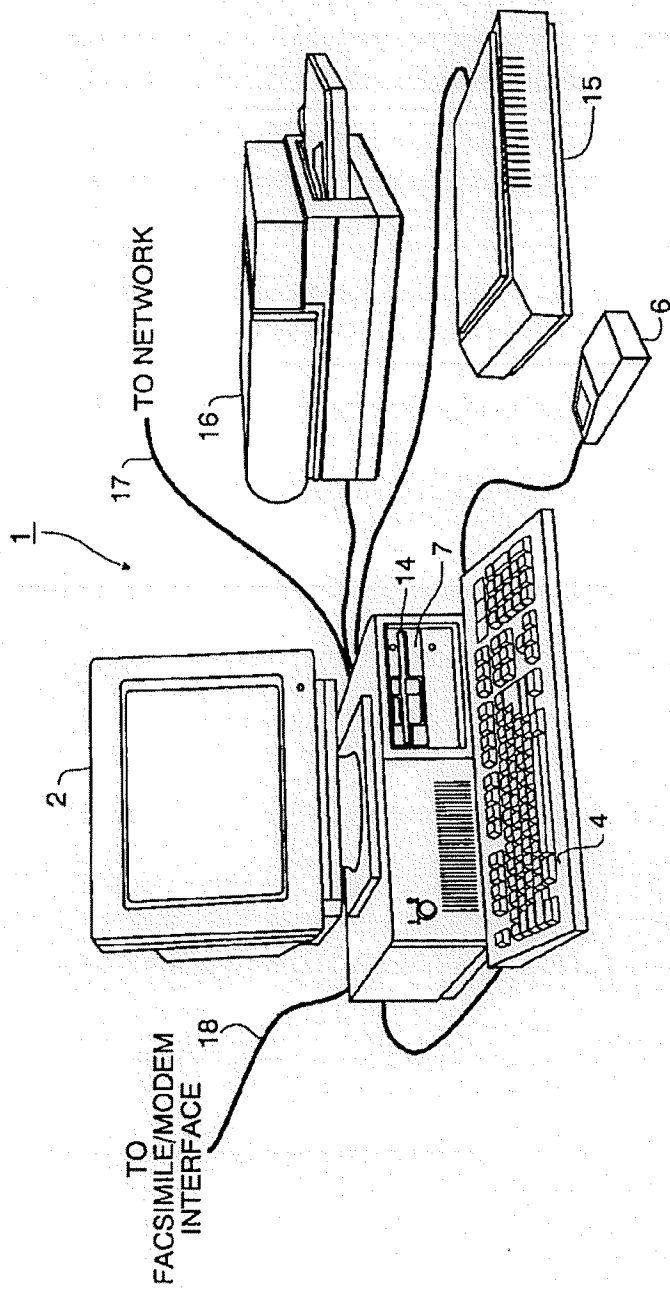


FIG. 1

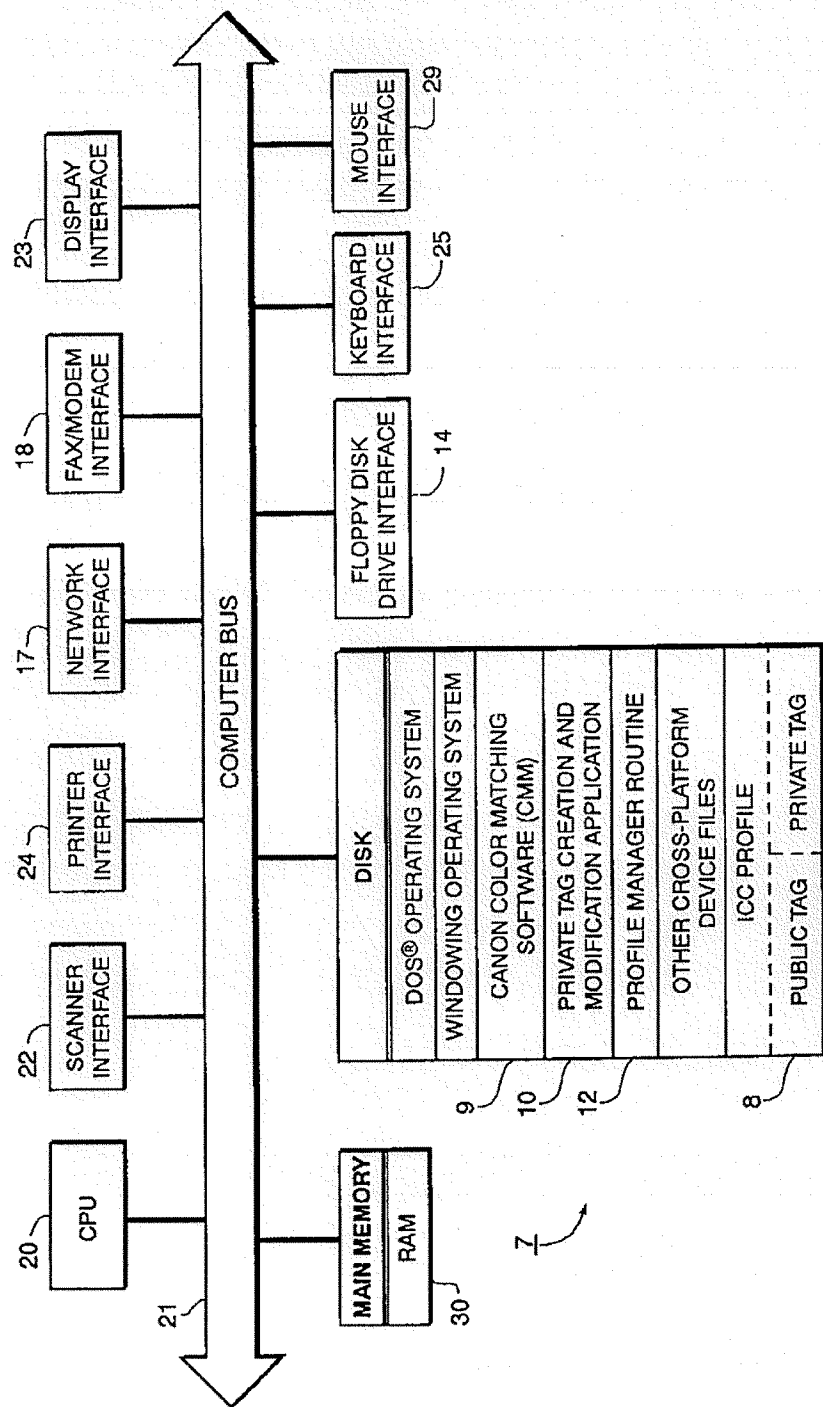


FIG. 2

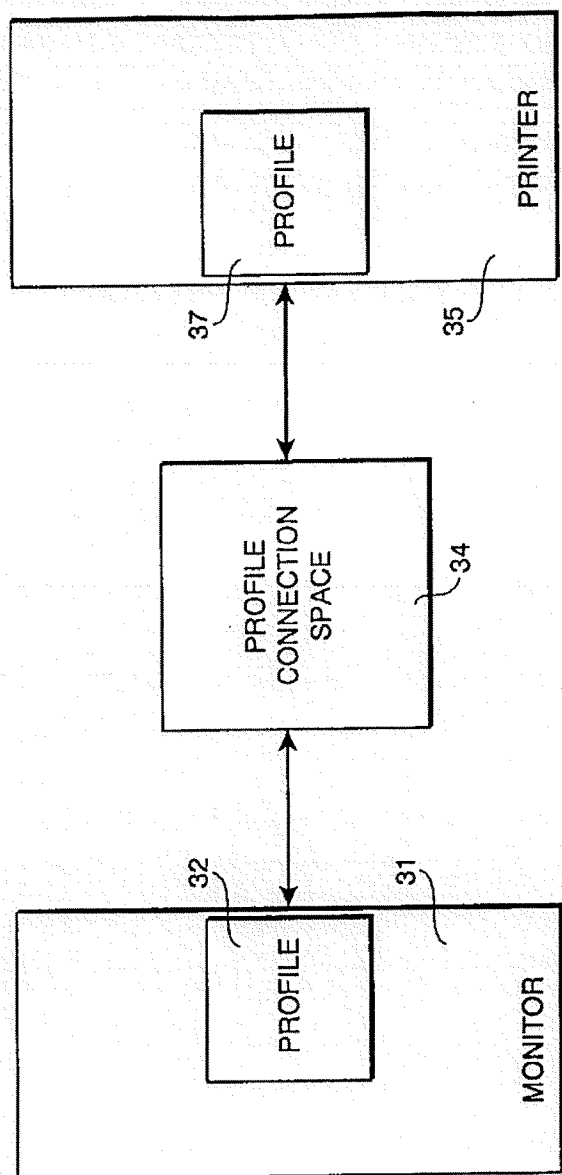


FIG. 3



```
size: 259354 bytes
CMType: "UCCM"
version: 0x02000000
profileClass: "ptr"
dataColorSpace: "CMY"
interchangeSpace: "Lab"
CreationDate: 23.6.95,
CS2Signature: "acsp"
```

Taq Table: (19 elements, double-click to inspect)

Ind	Signal	elementOffset	size
0	'meas'	360 = 0x00000168	20
1	'cpri'	380 = 0x0000017C	82
2	'dmnd'	462 = 0x000001CE	115
3	'dmdd'	577 = 0x00000241	123
4	'wldt'	700 = 0x000002BC	20
5	'tech'	720 = 0x000002D0	12
6	'desc'	732 = 0x000002DC	147
7	'pre0'	879 = 0x0000036F	109395
8	'pre1'	879 = 0x0000036F	109395
9	'pre2'	879 = 0x0000036F	109395
10	'gamt'	110274 = 0x0001AEC2	37009
11	'B2A0'	147283 = 0x00023F53	109395
12	'B2A1'	147283 = 0x00023F53	109395
13	'B2A2'	147283 = 0x00023F53	109395
14	'A2B0'	147283 = 0x00023F53	109395
15	'A2B1'	147283 = 0x00023F53	109395

FIG. 4

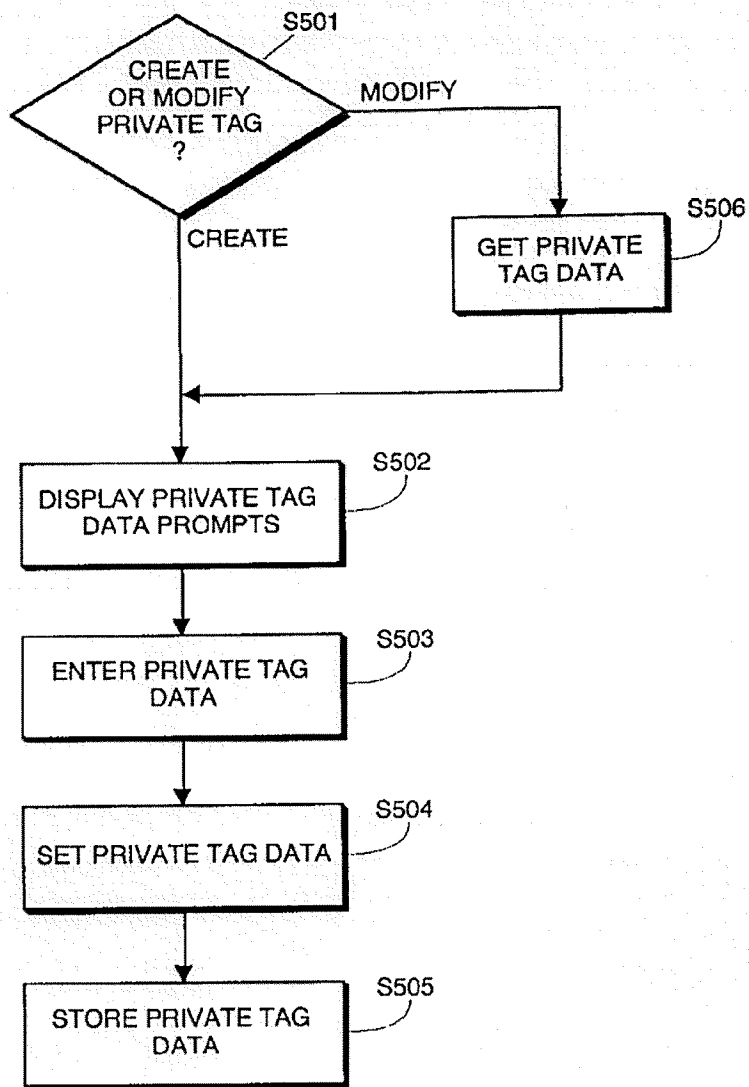
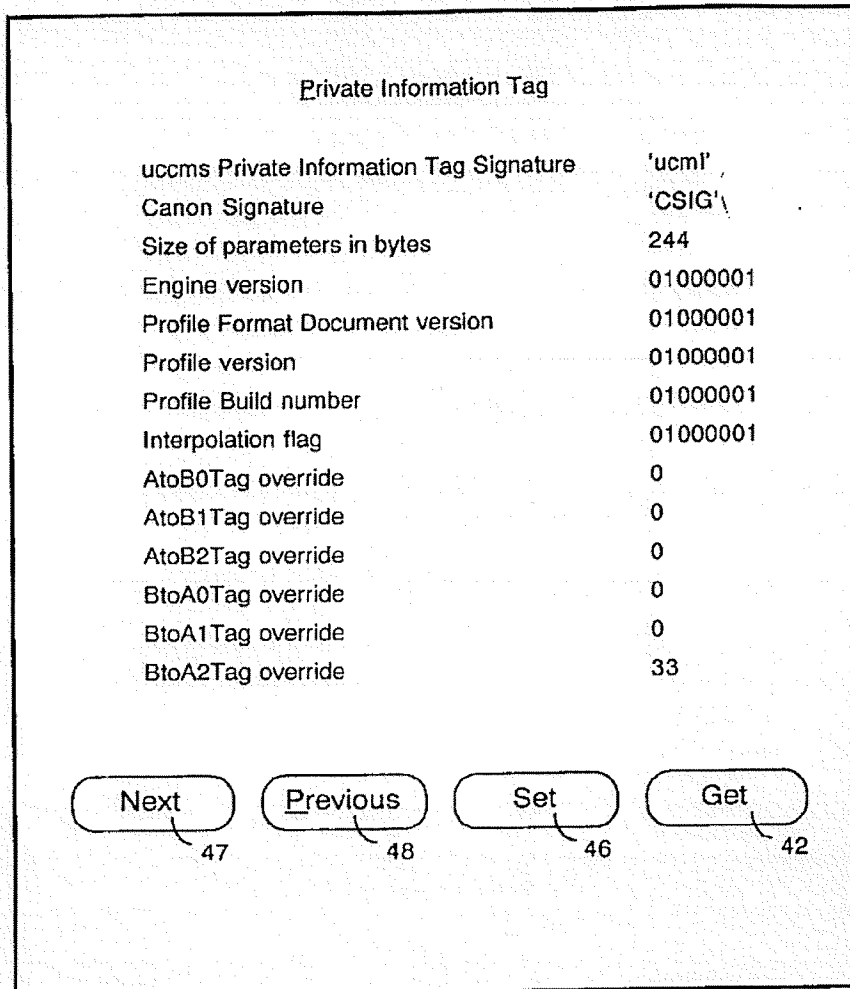
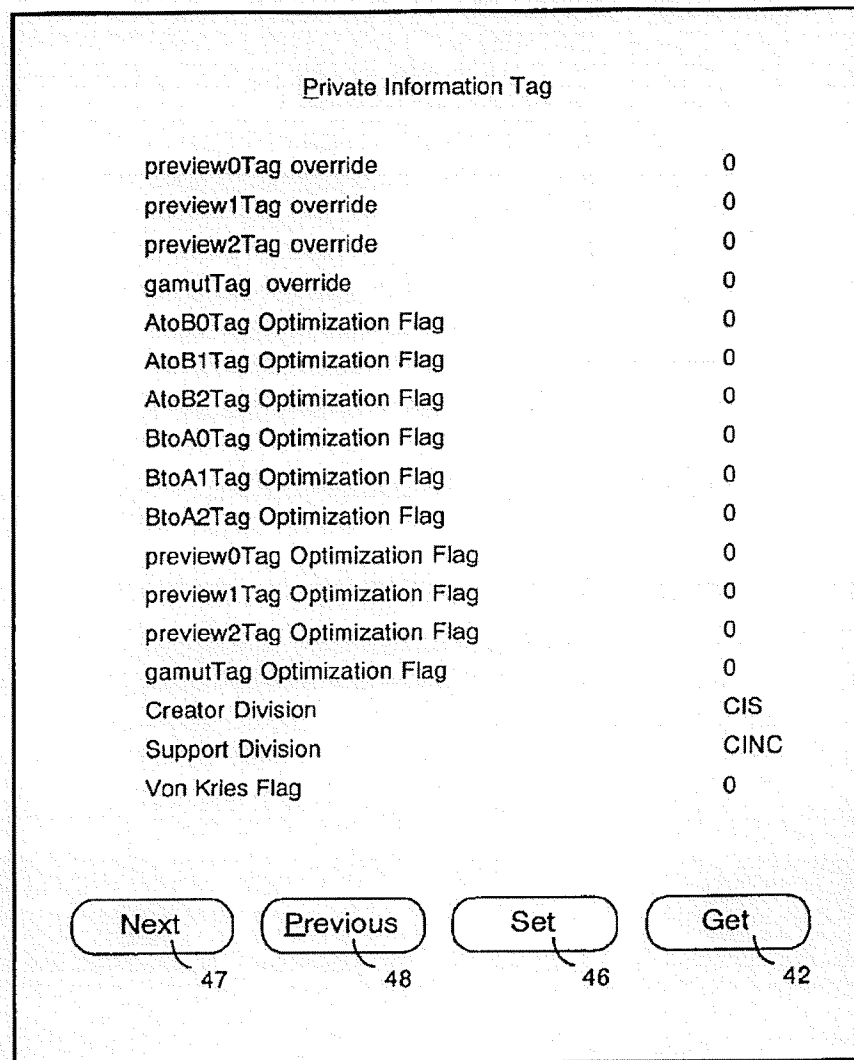


FIG. 5



43

FIG. 6



44

FIG. 7

byte(s)	Description	Data	Data Type
0-3	uccmsPrivateInformationTag Signature	'ucml'	icSignature
4-7	Canon Signature	'CSIG'	uint32
8-11	Size of parameters in bytes	244*	uint32
12-15	Engine version	0x01000001*	uint32
16-19	Profile Format Document version	0x01010000*	uint32
20-23	Profile version	0x01000000*	uint32
24-27	Profile Build number	0x00000001*	uint32
28-31	Interpolation flag	0x00000001*	uint32
32-35	AtoB0Tag override	0x0*	uint32
36-39	AtoB1Tag override	0x0*	uint32
40-43	AtoB2Tag override	0x0*	uint32
44-47	BtoA0Tag override	0x0*	uint32
48-51	BtoA1Tag override	0x0*	uint32
52-55	BtoA2Tag override	0x33*	uint32
56-59	preview0Tag override	0x0*	uint32
60-63	preview1Tag override	0x0*	uint32
64-67	preview2Tag override	0x0*	uint32
68-71	gamutTag override	0x0*	uint32
72-75	AtoB0Tag Optimization Flag	0x0*	uint32
76-79	AtoB1Tag Optimization Flag	0x0*	uint32
80-83	AtoB2Tag Optimization Flag	0x0*	uint32
84-87	BtoA0Tag Optimization Flag	0x0*	uint32
88-91	BtoA1Tag Optimization Flag	0x0*	uint32
92-95	BtoA2Tag Optimization Flag	0x0*	uint32
96-99	preview0Tag Optimization Flag	0x0*	uint32
100-103	preview1Tag Optimization Flag	0x0*	uint32
104-107	preview2Tag Optimization Flag	0x0*	uint32
108-111	gamutTag Optimization Flag	0x0*	uint32
112-175	Creator Division	CIS*	char[64]
176-239	Support Division	CINC	char[64]
240-243	Von Kries Flag	0x0*	uint32
...	Possible fields in the future		

FIG. 8

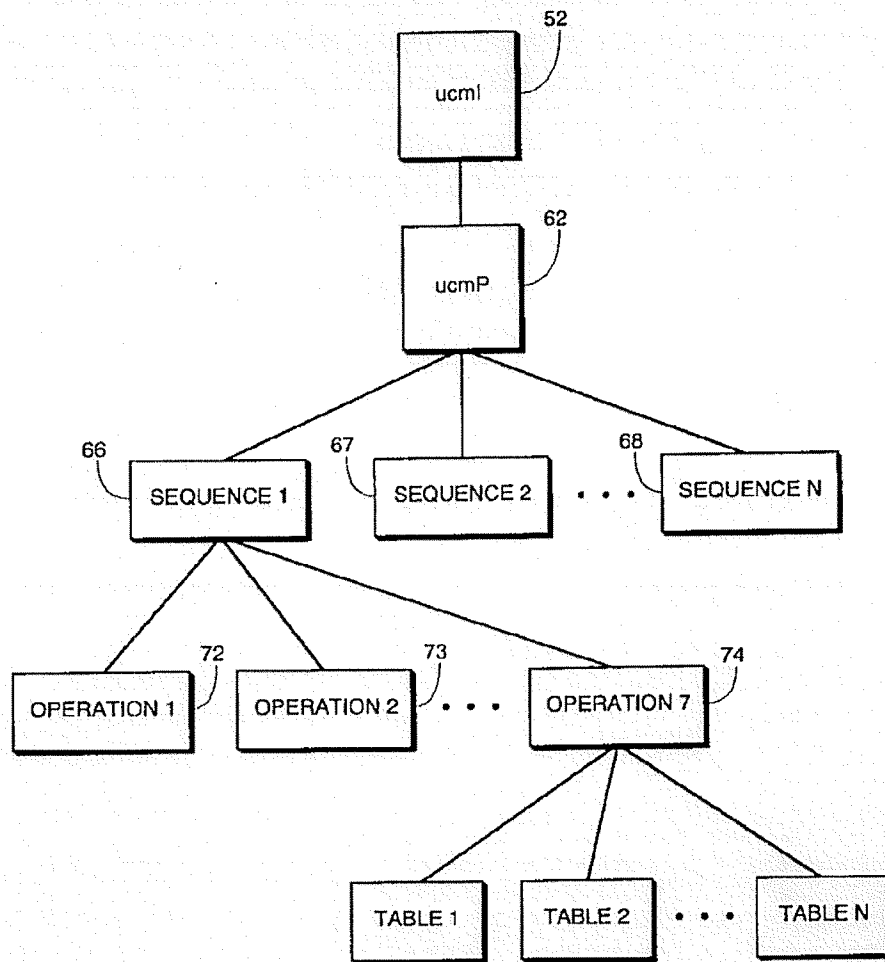


FIG. 9

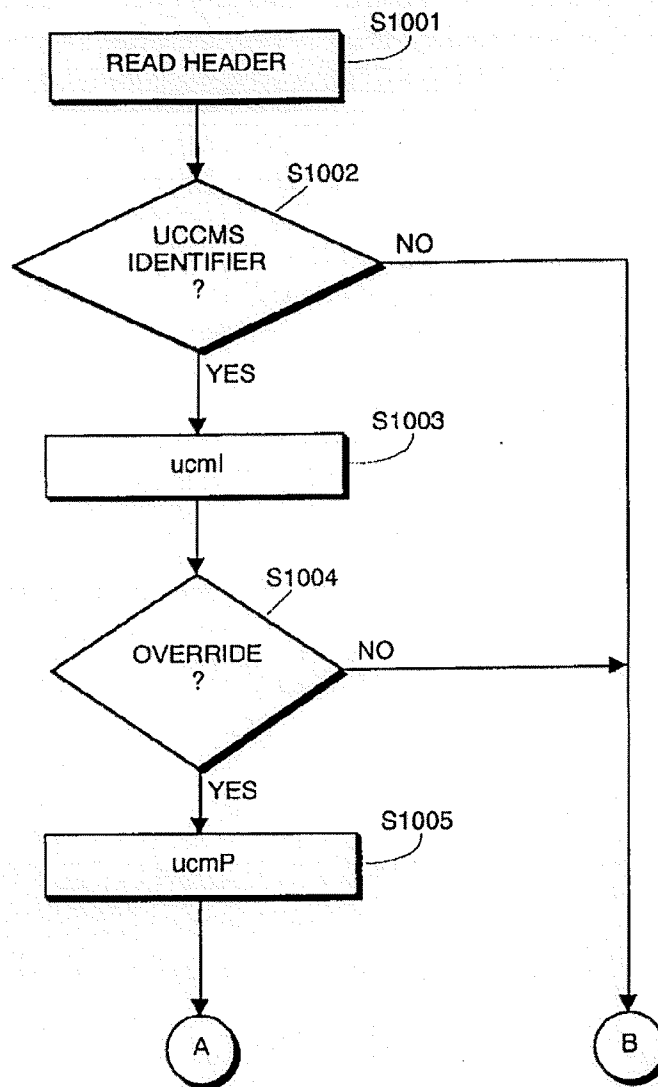
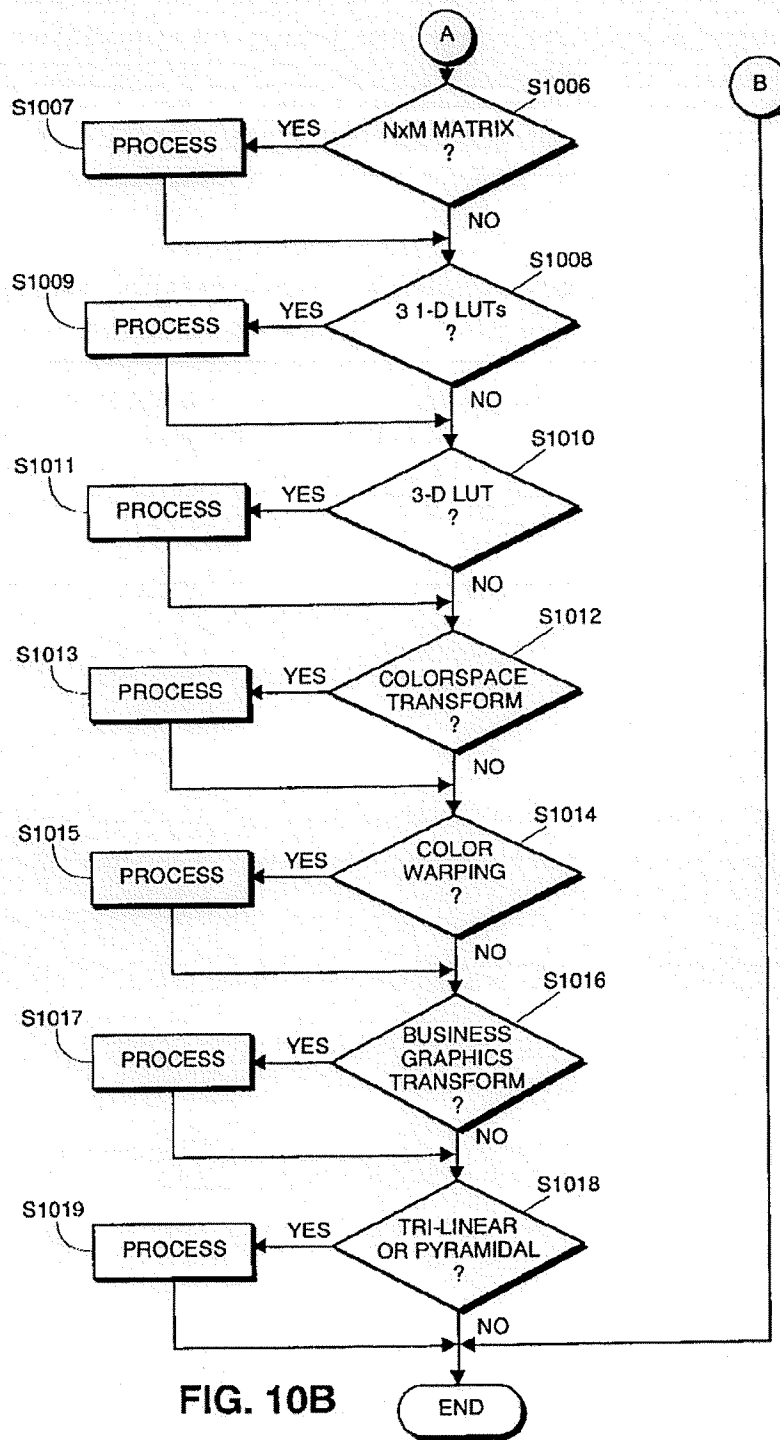


FIG. 10A



byte(s)	Description	Data	Data Type
0-3	uccmsPrivateSequencesTagTable Signature	'ucmP'	icSignature
4-7	Canon Signature	'CSIG'	uint32
8-11	Number of Sequences	4*	uint32
12-15	reserved	0x00	uint32
16-39	Sequences Structure 1*		uccmsSequenceStructure
40-63	Sequences Structure 2*		uccmsSequenceStructure
64-87	Sequences Structure 3*		uccmsSequenceStructure
88-111	Sequences Structure 4*		uccmsSequenceStructure

FIG. 11

```

/*UCCMS Sequence Structure*/
typedef struct {
uint32      33      CanonID; /*UCCMS ID for sequence structure*/
uint32      00      reserved; /*always put 0 here*/
uint32      04      numOps; /*number of operations in sequence*/
uint32      LAB      seqPCS; /*profile connection space of this sequence*/
uint32      40      length; /*length of operation tag in bytes*/
uint32      5163     offset; /*offset to operation tag*/
                                /*from byte 0 in 'ucmP'*/

```

FIG. 12


```

/*UCCMS Operation Structure*/
typedef struct {
uint32          oper;          /*UCCMS operation number*/
uint32          subid;        /*UCCMS sub ID number*/
uint32          length;       /*length of parameter list*/
                                   /*in bytes*/
uint32          tagFlag;      /*flag, if 0x00 use parm*/
uint32          parm[icAny]   /*as data, else use tag*/
                                   /*parameter list*/

```

FIG 13

byte(s)	Description	Data	Data Type
0-3	Operation ID	1	uint32
4-7	Sub ID	0x00	uint32
8-11	Length of Parameter List in bytes	44*	uint32
12-15	Tag Flag - Always use param data	0x00000000	uint32
16-	Parameter Data - n dimension	0x03*	uint32
	Parameter Data - m dimension	0x03*	uint32
	Parameter Data - Matrix Element a1	1*	s15Fixed16
	Parameter Data - Matrix Element a2	0*	s15Fixed16
	Parameter Data - Matrix Element a3	0*	s15Fixed16
	Parameter Data - Matrix Element b1	0*	s15Fixed16
	Parameter Data - Matrix Element b2	1*	s15Fixed16
	Parameter Data - Matrix Element b3	0*	s15Fixed16
	Parameter Data - Matrix Element c1	0*	s15Fixed16
	Parameter Data - Matrix Element c2	0*	s15Fixed16
	Parameter Data - Matrix Element c3	1*	s15Fixed16

FIG. 14

80

byte(s)	Description	Data	Data Type
0-3	Operation ID	2	uint32
4-7	Sub ID	0*	uint32
8-11	Length of Parameter List	0	uint32
12-15	Tag Flag	'A2B2'*	uint32

81 ↗

FIG. 15A

byte(s)	Description	Data	Data Type
0-3	Operation ID	2	uint32
4-7	Sub ID	0	uint32
8-11	Length of Parameter List	771*	uint32
12-15	Tag Flag	0	uint32
16	Parameter Data - dimension	0x03*	uint8
17	Parameter Data - # of input bits	0x08*	uint8
18	Parameter Data - # of output bits	0x08*	uint8
19-	Parameter Data - 1D table	0xFF*	uint8
	Parameter Data

82 ↗

FIG. 15B

byte(s)	Description	Data	Data Type
0-3	Operation ID	3	uint32
4-7	Sub ID	0	uint32
8-11	Length of Parameter List	1	uint32
12-15	Tag Flag	0	uint32
16-19	Table ID	0x11*	uint32

83 ↗

FIG. 16

byte(s)	Description	Data	Data Type
0-3	Operation ID	4	uint32
4-7	Sub ID	2*	uint32
8-11	Length of Parameter List	0*	uint32
12-15	Tag Flag	0x00	uint32

84 ↗

FIG. 17

byte(s)	Description	Data	Data Type
0-3	Operation ID	5	uint32
4-7	Sub ID	0	uint32
8-11	Length of Parameter List	6	uint32
12-15	Tag Flag	0x00	uint32
16-	Red printer color (m+y) hue angle	20*	uint32
	Green printer color (c+y) hue angle	175*	uint32
	Blue printer color hue (m+c) angle	300*	uint32

88 ↗

FIG. 18

byte(s)	Description	Data	Data Type
0-3	Operation ID	6	uint32
4-7	Sub ID	0*	uint32
8-11	Length of Parameter List	0	uint32
12-15	Tag Flag	'A2B2'*	uint32

89 ↗

FIG. 19A

byte(s)	Description	Data	Data Type
0-3	Operation ID	6	uint32
4-7	Sub ID	0	uint32
8-11	Length of Parameter List	771*	uint32
12-15	Tag Flag	0	uint32
16	Parameter Data - dimension	0x03*	uint8
17	Parameter Data - # of input bits	0x08*	uint8
18	Parameter Data - # of output bits	0x08*	uint8
19-	Parameter Data - 1D table	0xFF*	uint8
	Parameter Data

90 ↗

FIG. 19B

byte(s)	Description	Data	Data Type
0-3	Operation ID	7	uint32
4-7	Sub ID	0	uint32
8-11	Length of Parameter List	1	uint32
12-15	Tag Flag	0	uint32
16	Parameter Data -	0x08*	uint8
17	Parameter Data -	0x08*	uint8
18	Parameter Data -	0x08*	uint8
19-	Parameter Data -	0x08*	uint8
	Parameter Data

92 ↗

FIG. 20

byte(s)	Description	Data	Data Type
0-3	Operation ID	7	uint32
4-7	Sub ID	0	uint32
8-11	Length of Parameter List	1	uint32
12-15	Tag Flag	0	uint32
16-19	Table ID	0x04*	uint32

93

FIG. 21

byte(s)	Description	Data	Data Type
0-3	uccmsPrivateInterpolation Table Tag Signature	'ucmT'	icSignature
4-7	Canon Signature	'CSIG'	uint32
8-11	Number of Tables	3*	uint32
12-15	Reserved	0	uint32
16-	Table 1		uccmsIntrpLUTType
	Table 2		uccmsIntrpLUTType
	Table 3		uccmsIntrpLUTType

94

FIG. 22

```

/*UCCMS Interpolation Table Header Structure*/
typedef struct {
    uint32    tableID;        /*ID number of table*/
    uint32    length;        /*length of table in bytes*/
    uint32    offset;        /*offset to table from*/
                                /*byte 0 of ucmT*/

```

FIG. 23

Byte(s)	Description	Data	Type
0-3	tableKind	0*	uint32
4-7	parentTag	0x00*	icSignature
8	inChannels	0x03*	uint8
9	inBits	0x08*	uint8
10	outChannels	0x04*	uint8
11	outBits	0x08*	uint8
12	gridPoints3DTable	0x21*	uint8
13	gridPoints4DTable	0x21*	uint8
14-15	reserved	0x00	uint16
16-	CLUT		

96

FIG. 24

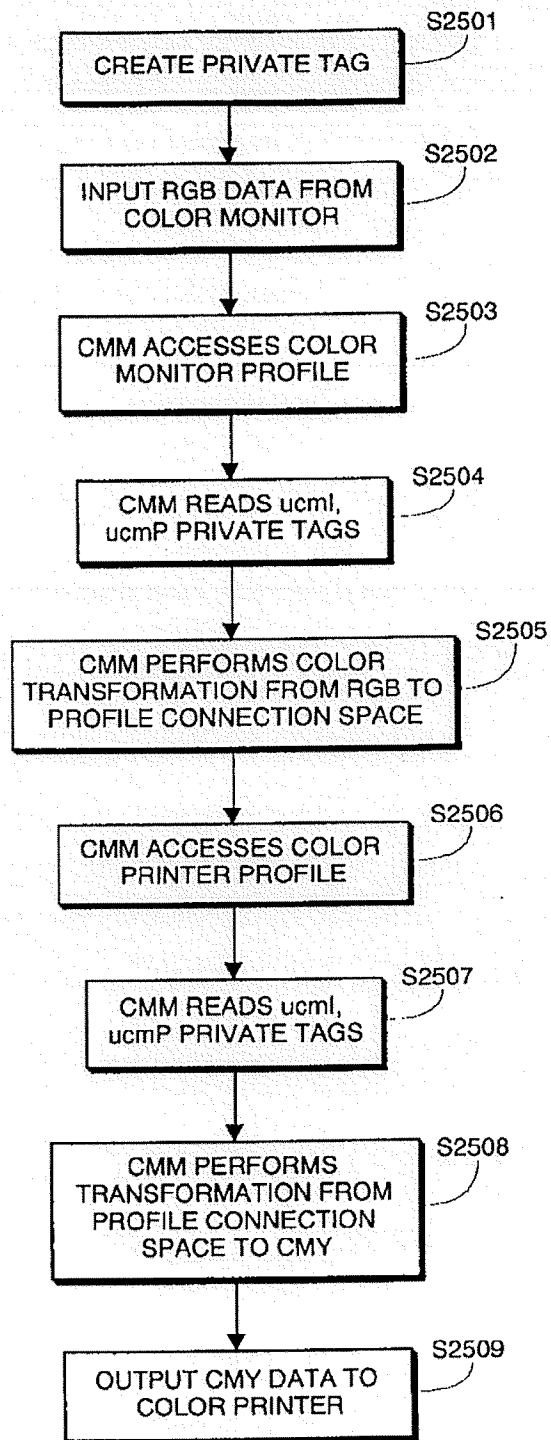


FIG. 25

1. Abstract

A system of performing a color transformation operation on input image data, the system using a first tag and a second tag to perform the color transformation operation in place of a predetermined series of color transformation operations stored in a public tag in a profile format. The system includes an inputting step for inputting color image data, a storing step for storing a first tag and a second tag, the first tag for storing override information for overriding the predetermined series of color transformations, and the second tag for storing color transformation operation data accessible via a hierarchical storage structure, and a determining step for determining, based on the override information in the first tag, whether to access the second tag. Also included in the system are a reading step for reading the color transformation operation data in the second tag in a case that the determining step determines to access the second tag, the reading step following pointers within the hierarchical storage structure to access the color transformation operation data in the second tag, and a processing step for performing a color transformation operation on the input image data in accordance with the color transformation operation data read in the reading step.

2. Representative Drawings

Fig. 25